[redacted]

DRA

(NASA-CR-120377) OPERATORS MANUAL FOR
THE MAGNETOGRAPH PROGRAM (SECTION 2)
Final Report (Lockheed Missiles and Space
Co.) 73 p HC $6.75
CSCL 14B

N74-30892

Unclas
G3/14 17092

Lockheed

MISSILES & SPACE COMPANY, INC.

A SUBSIDIARY OF LOCKHEED AIRCRAFT CORPORATION

SUNNYVALE, CA

13

LMSC D402244

FINAL REPORT

Section II

OPERATORS MANUAL
FOR
THE MAGNETOGRAPH PROGRAM

May 1974

Prepared for

George C. Marshall Space Flight Center
Huntsville, Alabama 35812

Contract No. NAS8-28018

Principal Investigator: Dr. A. M. Title

Lockheed Solar Observatory
Lockheed Palo Alto Research Laboratory
3251 Hanover Street
Palo Alto, California 94304

OPERATOR'S MANUAL FOR MAGNETOGRAPH PROGRAM

Larry November, August 21, 1973

To generate a directional magnetogram from original Spectra-Spectroheliograph data is a four pass process. Using the microdensitometer PASS1 scans the film to produce disk files numbered sequentially which contain the raw scan data. PASS2 may be used at any time in the process (i.e. when the disk gets full) to organize the data into a workable form and put this data on magtape. After all the data has been put on magtape PASS3 can be started which determines the field strength, angle, and velocities at each point on the sun. Near the end of the PASS3 run the computer requests a new magtape to output the final magnetograph. PASS4 can be used to examine the magnetograph final tape to produce any of eight types of magnetogram pictures (using the microdensitometer) or for types of vector magnetograms(using the Tektronix display scan).

This manual for use on the magnetograph program describes; 1. black box use of the program, 2. the magtape data formats used, 3. the adjustable control parameters in the program, and 4. the algorithms. With no adjustments on the control parameters this program may be used purely as a black box. For optimal use, however, the control parameters may be varied. The magtape data formats are of use in adopting other programs to look at raw data or final magnetograph data. For completeness I have included elaborate descriptions of how things work.

The sections of this manual are ordered as follows:

4. Algorithms

4.1 PASS2

4.2 PASS3

4.3 PASS4

1. Using the Program - A Typical Run

Given spectra-spectroheliograph data covering a region on the sun, information is available to readily determine velocities, B field strengths, and B field angles. This section describes a four pass computer process which is capable of deriving this information from the raw data. The sub-sections of this part of the manual are sequentially ordered to present the complete process of how to make a final vector magnetogram from the raw film data.

1.1 About the Film - IX Coordinate and Polarization Pair

The spectra-spectroheliograph film has two basic formats, the old type, and the new type, which are shown in Figure 1.1. The basic characteristics of these two types are different and it is important when running PASS1 to distinguish between them.

A COMPLETE RUN is composed of several REGION SCANS which are each composed of many SLIT PAIRS. Before each REGION SCAN the spectra-spectroheliograph was pointed at some initial solar coordinate and allowed to step across the region making many SLIT PAIRS each at a different X coordinate. After a change in polaroids the next REGION SCAN was begun. In each REGION SCAN the same pattern of spots was crossed and a similar looking sequence of SLIT PAIRS encountered.

It is extremely important that the film be correctly identified or the analysis cannot correctly proceed. Each SLIT PAIR is identified by an X coordinate integer which I refer to as IX. IX sequentially increases as each SLIT PAIR in a REGION SCAN is crossed. Also IX is the same for SLIT PAIRS that were taken at the same solar coordinate in separate REGION SCAN's of the same COMPLETE RUN. It is essential that (1) IX increase sequentially and (2) that an accurate corresponse exist between the SLIT PAIR numbering and the actual solar coordinates.

Sometimes only a portion of a REGION SCAN needs be analyzed which places one more restriction on IX. I introduce the term ANALYZING SCAN to represent that portion of the REGION SCAN that is analyzed. Just as there may be four REGION SCANS for a COMPLETE RUN there may be (at most) four ANALYZING SCANS. Figure 1.2 portrays how the ANALYZING SCAN fits in. Note that each ANALYZING SCAN in a COMPLETE RUN contains the same set of IX coordinates. Also note that the first SCAN PAIR of the ANALYZING SCAN is labeled IX=1. This is the third and last restriction on IX.

In addition to IX each SCAN PAIR has another characteristic associated with it. This characteristic is its polarization. Each pair physically represents the light in two mutually orthogonal polarizations. Thus a SCAN APAIR may represent RHC-LHC polarizations, or linear 37° - linear 127°, a linear 0° - linear -90°. Each REGION SCAN has associated with it one such polarization pair, and all the SCAN PAIRS in that REGION SCAN WERE TAKEN through the same pair of polaroids.

Each SCAN PAIR can thus be uniquely identified by IX and its polarization pair.

1.2 Densitometer Setup

Densitometer Setup requires two operations. First the film must be properly cleaned and placed on the platten. Second the microdensitometer must be calibrated and focused for the film.

The film should be cleaned and placed on the platten such that the REGION SCAN begins on the right and ends somewhere in the roll on the left. IX = 1 should be placed somewhere near the right edge of the platten with IX increasing to the left (as the densitometer is faced). The film must be aligned such that the film edge lies precisely parallel to the microdensitometer X axis. This can be checked by running the densitometer back and forth in manual.

Once aligned the densitometer may be precisely focused, a small aperture installed and aligned, and the transmission set for a clear portion of the film. The exact procedure is not described here as an elaborate method is given in the 'Microdensitometer System' by Steve Schoolman.

1.3 Initializing the Computer System

Using the program PIP on the PDP-11 system initialization of the DISK and the MAGTAPE can be accomplished. A SCRATCH disk should be installed and running on DKØ, and a blank tape running on MTØ.

Five restrictions exist for the scratch disk. First, at least 3000 blocks of free disk must be freed. Second, a file called NUMBER must be deleted from the user's directory. Third, a file called PARM must be deleted from the user's directory. All these conditions can be met using PIP. Third, the file PASS1.LDA must be present on the disk. Fourth, the file PASS2.LDA must be present on the disk. Both of these files can be found on MAGTAPE S-24 under the User Initials [2,11] (please do not zero this tape).

It is necessary that a ZERO mark be present on the output tape. If one is, no more action need be taken. If one is not the command MTØ=/ZE in PIP will zero the tape (and wipe out anything already on the tape).

Once both the disk and tape are readied the program is ready to begin.

1.4 A Warning

The program PASS2 has defined within a BLOCK DATA statement all the program control parameters. In general those parameters should work for any piece of film. However, a new dimension in film size or raw step wedge data could lead to trouble. It may be necessary under unusual conditions to alter the control parameters and re-link PASS2 Section 3 of this booklet describes the conditions and the procedures under which PASS2 should be altered.

1.5 Zeroing the Microdensitometer

Zeroing is accomplished with the program PASS1.LDA which should be called in from disk and running. PASS1 is just the system program TRACE with one alteration - it smooths data as it scans which TRACE does not do. PASS1 (and TRACE) is a user interaction program with many features only a few of which I describe here. PASS1 is in idle and waiting for a user response if a * is the last display character.

With the densitometer in MANUEL the densitometer γ should be moved to STARTING POSITION 1 on the film (see Figure 1.2). Z is the command which will then zero the reading to this starting point.

1.6 Labeling the Output File

The command I in PASS1 causes the computer to respond with:

IDENT:

After which the user can enter a label of 31 characters length or less. A typical label identifies the region and the time of the COMPLETE RUN.

1.7 User Defined Input Parameters

Essentially three decisions are now left up to the user. He may select the parameter L = length of film he wants to scan (microns), ΔX = the resolution

needed in X (microns), and ΔY = the resolution reading in Y (microns). Typically these parameters are on the order of L = 15cm = 150,000 μ, ΔX = 15μ, ΔY = 200μ. Having decided on these parameters the user-computer dialogue should proceed as follows (computer response is underlined):

For old data:

   * U

      USER-DEFINED SCAN PARAMETERS:

      X-DIR= +

      Y-DIR= -

      PATTERN= R or E (user's choice)

      DELTA X= value of ΔX <CR>

      POINTS/LINE= value of $L/\Delta X$ ($\leq$24,576) <CR>

      YSTEP= value of ΔY <CR>

      LINES= value of $57150/\Delta Y$ <CR>

      X= 0 <CR>

      Y= 0 <CR>

      (remember the line feed termination character)

For new data:

   * U

      USER-DEFINED SCAN PARAMETERS:

      X-DIR= +

      Y-DIR= -

      PATTERN= R or E (user's choice)

DELTA X=   value of ΔX <CR>

POINTS/LINE= value of $L/\Delta X$ ($\leq$24,576) <CR>

YSTEP=   value of ΔY <CR>

LINES=   value of 30,000./ΔY <CR>

X= Ø <CR>

Y= Ø <CR>

X= Ø <CR>

Y= -30480 <LF>

(remember the line feed).

If errors are made on this input it may be corrected by typing U again - this time the computer will respond:

*   U

OPTION:

Correction may be made to any of the parameters by selecting the appropriate option. These are the options:

U - complete dialogue

X - X direction

Y - Y direction

P - Pattern

C - Starting coordinates

DX - Delta X

DY - Y step

NP - Number of points per line

NL - Number of lines

E - exit to monitor from PASS1.

1.8 The DENSITOMETER SCAN

Now the actual DENSITOMETER SCAN can begin. The intiating character for the scan is S. PASS1 first opens a file for output of the data on disk, and then proceeds to scan recording the data on disk. At the end of the scan a * flashes on the display screen and the process may continue.

1.9 PASS2

Generally speaking one densitometer scan is enough to fill up the disk. PASS2 should therefore be called after each densitometer scan to move the data onto tape.

PASS2 first will ask to know which files it is to work on off of disk. This is specified by the letter which identifies the file on disk and the number of the first and last file to be analyzed from disk (these two numbers may be equal if only one file is to be analyzed).

When PASS2 is brought into core for the first time during a COMPLETE RUN it must be initialized with certain parameters. It will ask for the following parameters:

WAVELENGTH SCALE (MICRONS WAVELENGTH/MICRON ON FILM)

($\approx$1.4ØØØØE-6)

XSTEP (SECONDS OF ARC /SLIT PAIR) ($\approx$0.5)

YUNIT (SECONDS OF ARC/MICRON ON FILM) (≈2.40380E-3)

WAVELENGTH OF FILM LEADING EDGE (MICRONS) (≈52.4994)

WAVELENGTH OF LINE CENTER (MICRONS) (≈52.502)

WAVELENGTH OF ANY CALIBRATION LINES (NONE - <CR>)

Next PASS2 will tell which data file it has opened and request input of the initial IX value for the densitometer scan. IX=1 for the first scan or in general it is equal to the value of IX for the first SLIT PAIR in the DENSITOMETER SCAN.

PASS2 will next request the two orthogonal polarizations of the SLIT PAIRs in this DENSITOMETER SCAN. The information for RHC polarization is coded 511, and for LHC polarization 256. Linear polarization is entered by the angle of the polaroid in degrees from the X-scan direction on the sun.

PASS2 will then proceed to move this data onto tape in a more convenient format. As a warning PASS2 requires 100 contiguous scratch disk blocks. If it cannot allogate this space a STOP 000001 will occur. This situation can be remedied by using the /PK switch in PIP or by deleting more files on disk and rerunning PASS2. PASS2 takes approximately 15 minutes to run through a full DENSITOMETER SCAN file.

1.10 The Next DENSITOMETER SCAN

Once PASS2 and PASS1 have run the files PARM and NUMBER should be present on disk. Do not delete either of these two files until the end of the COMPLETE RUN. Once again, though, room must be made on disk which can be accomplished by deleting the old data file. Once the disk is readied the next DENSITOMETER SCAN may begin.

Move the film on the PLATTEN to the next starting point on the film (see Figure 1.2). Once again the Densitometer must be properly calibrated and the film set parallel to the Densitometer X axis (see Section 1.2). Once done, PASS1 must be rerun and zeroing (section 1.5), labeling (section 1.6), and initialization of input parameters (section 1.7) redone. Please note that the input parameters L and ΔX may vary from DENSITOMETER SCAN to DENSITOMETER SCAN, but that ΔY must not change. (as L, ΔX, ΔY are defined in section 1.7). After all is readied S (see section 1.8) initializes the DENSITOMETER SCAN.

After the SCAN PASS2 should be run. This time through, PASS2 will ask only for the data file letter and number, IX for the first SLIT PAIR in this DENSITOMETER SCAN, and for the orthogonal polarizations of this DENSITOMETER SCAN. After PASS2 finishes this DENSITOMETER SCAN will have been completely moved to tape and the next DENSITOMETER SCAN begun.

1.11 After the Data Tape is Done

After the last of the DENSITOMETER SCANS in the COMPLETE RUN has been moved to tape by PASS2 the actual magnetic field information can be processed. PASS3 performs this function.

PASS3 is recorded on tape S-24 under the UIC (2,11) and must be moved to disk before being run. Three considerations should be made before running PASS3. First, for optimal running the control parameters should be correctly adjusted (see section 3). Second, PASS3 requires (number of points in X) *number of points in Y)/64 blocks of contiguous scratch area on disk which must be available. Third a tape with a correct zero mark should be ready and waiting for output of the final magnetogram.

When the disk is ready and the data tape is mounted on MTØ and moved to the load point then PASS3 may be run. PASS3 first requests the maximum picture dimensions and then a single letter which defines the files on the data tape to be processed. PASS3 then proceeds to process the data which takes approximately (Number points in X)*(Number of Points in Y)/200 minutes.

At the completion of the data processing, PASS3 will ring a bell and request the final magnetogram tape be mounted on unit MTØ. Once loaded and an output file is assigned to unit 8, PASS3 should be continued. After output of the magnetogram PASS3 will rewind the tape and return control to the monitor.

1.12 PASS4

Final display of the data from the magnetogram tape in various formats is possible using PASS4. PASS4 requires only that the magnetogram tape be mounted on MTØ and that sufficient space is available on disk for output of the picture.

When PASS4 is run it first will request assignment of the magnetogram file to tape unit 8. After a continuation is given by the user PASS4 will request two integers, MODE and ITYPE. MODE=0 causes an exit to monitor. The other MODE and ITYPE possibilities are described:

MODE=1, ITYPE=1 - A dopplergram will be produced. The dopplergram is written on a disk file whose name is given by the program. This file may later be output on the densitometer as a picture.

MODE=1, ITYPE=2 - A magnetogram will be produced. The magnetogram is written on a disk file which may later be rendered as a photograph by the densitometer.

MODE=1, ITYPE=3 - A gamma-gram is produced. This disk file may later be rendered on the densitometer as a photograph which displays fields toward the observer a block (0° yields transmission of 0) and fields away from the observer as white (180° yields transmission of 1023).

MODE=1, ITYPE=4 - A phi-gram is produced. This disk file may later be rendered on the densitometer as a photograph which displays fields in the X scan direction as black (0° yields transmission of 0) and fields in the -X scan direction as white (180° yields transmission of 180°).

MODE =2, ITYPE=1 - A longitudinal magnetogram is produced. B cos (δ) is written on the output file which may later be rendered as a photograph.

MODE =2, ITYPE=2 - A transverse magnetogram is produced. B sin (δ) is written on the output file which may later be rendered as a photograph.

MODE =3, ITYPE=1 - Three longitudinal magnetograms are produced each representing a different color. B cos (δ) is written on the output files, and the color wheel red-blue-yellow represents the field angle of going from 0° to 180° from the X scan direction.

MODE=3, ITYPE=2 - Three transverse magnetograms are produced each representing a different color. B sin (δ) is written on the output files, and the color wheel red-blue-yellow represents the field angle ϕ going from 0° to 180° from the X scan direction.

MODE=4, ITYPE=1 - A vector magnetogram is displayed on the tektronix screen as seen from the viewing angle (δ, ϕ) as requested by the program. All fields are displayed which are greater then BMIN (program request) as vectors of equal length which point in the direction of the field and are anchored to the surface of the sun in the appropriate location. After completion of the drawing the bell is sounded. Striking a return on the keyboard will cause a hardcopy to be produced and the program to continue.

MODE=5, ITYPE=1 - A vector magnetogram is displayed on the tektronix screen as seen from the viewing angle (δ, ϕ) (program request). All fields are displayed which are greater than BMIN (program request) as vectors of relative length

which point in the direction of the field and are anchored to the surface of the sun in the appropriate location. After completion of the drawing the bell is sounded and a return from the keyboard will cause a hardcopy to be produced.

The files that have been created by PASS4 in modes 1 through 3 may be rendered as photographs by the densitometer using the system program TRACE. Setup of the densitometer is described in the "Operator's Manual for Microdensitometer Control Program' by Steve Schoolman. The film should be on the platten with the densitometer at the lower-left of the film. Playback of the photograph is initiated by the P command in running TRACE.

```
$ RUN TRACE
  MONITOR
* Z
* P
```

the computer then responds and the dialogue continues:

```
SCALE FACTOR  user's choice 1 to 100 <CR>
PARAMETER SOURCE?  R
FILE NAME:  complete name of file created by PASS4 <CR>
TYPE ANY KEY TO CONTINUE
```

With the lights out and the film placed on the platten the user may initiate the picture output by typing any character. Completion of the picture output is signified by a bell.

2. Data Formats

Often times the user may want to use the output data or the output magnetogram in ways that are not possible through this system. In order to allow for easy use of the output information I now describe the data formats which are used.

I describe here a total of four data formats. They are: 1) The format of the data at the end of PASS1 as it is stored on disk, 2) the format of the data tape as it is produced by PASS2, 3) the format of the magnetogram tape as produced by PASS3, and 4) the format of the picture records as produced by PASS4 in modes 1 through 3.

Each of these total mediums will contain one or many files each of which will contain many Fortran unformatted records. Each record must be read in a single Fortran READ statement and may contain as many as 12000 integer words (2 bytes each). All the records, in all the files, in all four storage media are of the same general format. The first word of each record is a code word, which describes the nature of the record, and the second word is a length word which tells how many words remain in the second. The rest of the record contains the desired information.

2.1 Disk Data Files after PASS1

Each disk data file contains microdensitometer data from one DENSITOMETER SCANS A COMPLETE RUN is liable to consist of many DENSITOMETER SCANS and so many of these data files.

The data file consists of many records. The first two or three of these records contains control information and then each additional record contains the raw densitometer data from each line of the scan, one record to each line. The record contains the following.

RECORD 1 = Label:

WORD 1 = ICODE = 1 or 4

WORD 2 = N

WORD 3 to WORD N+2 = a label 2N bytes long

RECORD 2 = Message (optional)

WORD 1 = ICODE = 2

WORD 2 = N

WORD 3 to WORD N+2 = a message 2N bytes long

RECORD 2 or 3 = Parameters

WORD 1 = ICODE = 3

WORD 2 = N = 141

WORD 3 = NPPL = Number of points per DENSITOMETER SCAN line.

WORD 6 = IDELX = distance between points in microns.

WORD 7 = IDELY = distance between lines in microns.

WORD 10 = NLPF = number of lines per frame (for old data this is the number of lines for new data this is half the number of lines).

WORD 11 = -1 for raster pattern, 0 for edge scan pattern.

WORD 14 = 0 for old data, 2 for new data.

RECORD 3+ or 4+ = Densitometer Data.

WORD 1 = ICODE = - line number

WORD 2 = N = NPPL

WORD 3 to WORD N+2 = the data.

More information is available in the parameter record than is defined here. A complete description can be found in the output Format section of the 'Operator's Manual to Trace' by S. Schoolman.

2.2 Data Tape Format After PASS2

The data tape produced by PASS2 contains a sequence of files labeled <letter> n.DAT where n varies from 1 to the total number of DENSITOMETER SCANS in the COMPLETE RUN. Each of these files contains the following sequence of records:

RECORD 1 = Label

WORD 1 = ICODE =1

WORD 2 = N

WORD 3 to WORD N+2 = label of length 2N bytes.

RECORD 2 = Parameters

WORD 1 = ICODE=2

WORD 2 = N = 10

WORD 3 and 4 = XSTEP = seconds of arc per step in scan direction

real number four bytes long).

WORD 5 & 6 = YSTEP = seconds of arc per step in Y (real).

WORD 7 & 8 = ZSTEP = microns wavelength between data points (real).

WORD 9 & 10 = ZCEN = microns wavelength of centerline (real).

WORD 11 = NY = number of steps in the Y direction.

WORD 12 = 1 if old data, 2 if new data.

RECORD 3+ = Data for one slit crossing of SLIT PAIR.

WORD 1 = ICODE = 0

WORD 2 = N

WORD 3 = J = number of densitometer points in this slit crossing.

WORD 4 = IX = X step number of SLIT PAIR.

WORD 5 = IY = Y step number.

WORD 6 & 7 = ZØ = number of steps from first densitometer point to line center (real).

WORD 8 = IPOL = polaroid for this slit crossing (RHC-511, LHC-256, linear integer angle in degrees).

WORD 9 to WORD J+8 = film transmission smoothed and photometered.

2.3 Data Tape Format for Magnetogram After PASS3

This tape has only one file for the COMPLETE RUN. Each record contains information on one coordinate point encoded as follows:

RECORD 1 = Label

WORD 1 = ICODE = 1

WORD 2 = N

WORD 3 to WORD N+2 = label of length 2N bytes

RECORD 2 = Parameters

WORD 1 = ICODE = 2

WORD 2 = N = 6

WORD 3 & 4 = XSTEP = seconds of arc per step in X (real, four bytes long).

WORD 5 & 6 = YSTEP = seconds of arc per step in Y (real).

WORD 7 = NX = total number of steps in X direction.

WORD 8 = NY = total number of steps in Y direction.

RECORD 3+ = Data

WORD 1 = ICODE = 0.

WORD 2 = N = 10

WORD 3 = IX = number of X step for this data point.

WORD 4 = IY = number of Y step for this data point.

WORD 5 & 6 = V = velocity (Km/sec, positive-downward) (real).

WORD 7 & 8 = B = field strength (gauss) (real).

WORD 9 & 10 = GAM = field angle in degrees from the line of sight (real).

WORD 11 & 12 = PHI = field angle in degrees from X (real).

2.4 Disk Picture Files from PASS4

Each picture is encoded on one file. The records are in the same format as those output by PASS1, i.e. capable of controlling the microdensitometer program TRACE:

RECORD 1 = Label

WORD 1 = ICODE = 4

WORD 2 = N

WORD 3 to WORD N+2 = label of length 2N bytes.

RECORD 2 = Message

WORD 1 = ICODE = 2

WORD 2 = N

WORD 3 to WORD N+2 = message.

RECORD 3 = Parameters

WORD 1 = ICODE =3

WORD 2 = N = 141

WORD 3 = NPPL = number of points per line.

WORD 6 = IDELX = number of microns between output points in X=10.

WORD 7 = IDELY = number of microns between output points in Y=10.

WORD 10 = NL = number of lines in output picture.

WORD 12 = ISPEED = 64

All other words in this record are zero.

RECORD 4+ = Data

WORD 1 = ICODE = -line number

WORD 2 = N

WORD 3 to WORD N+2 = transmissions of points on line.

3. Control Parameters and Program Relinking.

On tape S-24 all the necessary programs are stored for making a magnetogram. S-24 contains the following files under UIC [2,11]:

PASS1.LDA	PASS3A.FTN	DELETE.MAC
PASS2.LDA	PASS3B.FTN	
PASS3.LDA	PASS3C.FTN	
PASS4.LDA	PASS3D.FTN	
PASS2A.FTN	PASS4A.FTN	
PASS2B.FTN	PASS4B.FTN	
PASS2C.FTN	PASS4C.FTN	

Direct running can be accomplished using the default control parameters in the load modules. The control parameters for PASS2 are contained in PASS2C.FTN and the control parameters for PASS3 are contained in PASS3D.FTN. If these files are altered then PASS2 or PASS3 must be recompiled with the /ON switch and relinked. If PASS2C.FTN is changed PASS2 must be linked from PASSA/CC, PASS2B/CC, PASS 2E, and DELETE. If Pass 3D.FTN is altered then PASS 3 must be relinked from PASS3A/CC, PASS3B/CC, PASS3C/CC, PASS3D, DELETE.

Listings of these programs can be found at the end of this booklet. The listings are by file and give all the subprograms and programs used.

3.1 PASS2C.FTN

PASS2C is a BLOCK DATA statement which defines parameters in four common blocks. Each of these four commons is described with the parameters:

/FILM/ contains all the information on the film dimensions. In practice this block is the most critical and these parameters must be changed in switching program use from old to new data. Five parameters in /FILM/ are adjustable SLT(1), SLT(2), SMARG(1), SMARG(2), SEFF. Figure 3 illustrates these dimensions for the new and the old film. Each parameter is described below:

SLT(1)= size in microns of film of one slit crossing (default= old data = 2644.0)

SLT(2) = size in microns of film of second slit crossing in SLIT PAIR (default = old data = 2644.0)

SMARG(1) = distance in microns of film between second slit crossing in SLIT PAIR to first slit crossing of next pair (default = old data = 3527.0)

SMARG(2) = distance in microns of film between first slit crossing in SLIT PAIR to second slit crossing in next pair (default = old data = 685)

SEFF = distance around line center in microns wavelength which is useful - anything over this distance will be discoded (default = 1)

As a check 9500.0 microns represents the total recycling distance. No matter what value for SLT(1 & 2) and SMARG (1 & 2) are used this check applies.

On the new film format these four parameters have the values:

SLT(1) = 8500.

SLT(2) = 8500.

SMARG(1) = 1000.

SMARG(2) = 1000.

/DENS/ contains only one parameter ICONTR. The program in attempting to find the leading edge of the film looks for a slope in transmission of ICONTR over ten points. For old data it looks for a drop in transmission of ICONTR; for new data, it looks for a rise in transmission of this amount. If a STOP2 of the program results it is likely that this parameter was too large.

/WEDGE/ contains step wedge data if it is available. Two integer arrays are defined in /WEDGE/, ITRANS(11) and INT(11). ITRANS(n) represents the transmission reading that results for the corrected transmission INT(n). These two relations should be approximately linear, i.e. INT is approximately proportional to TRANS and ranges from about 0 to about 1000. NSTEP tells the number of useful steps in the wedge. If no correction is necessary or no wedge data is available then NOWEG =.TRUE. The default for this common is NOWEG=.TRUE.

/BUFFER/ has only one adjustable number - that is the size of the array IBUF. This number represents the maximum densitometer line length and is determined by the amount of core available on the machine. The default value of 12288 is about the maximum that will run in a 24K core machine.

3.2 PASS3D.FTN

PASS3D is a BLOCK DATA statement that contains program control parameters in four commons. These four commons are /PROFLE/, /LNPRM/, /FILTER/, /FFT/ and are described as follows:

/PROFLE/ contains two numbers which characterize the line profile. XWIDTH represents the half-height field width of the line in microns wavelength.

PKLIN represents the relative line depth ($\sim$25 x (continuum intensity)/(line center intensity)). The intensity profile is assumed to be approximately lorentzian with the following relative curve:

$$I = \frac{PKLIN}{4\left(\frac{\Delta\lambda}{XWIDTH}\right)^2 + 1}$$

The default values which the program uses are XWIDTH=.06E-2 microns wavelength and PKLIN=48.

/LNPRM/ contains other line parameters that apply to splitting and the doppler line shift. Only one control parameter is settable in this block; that is GFAC = Landé g factor for the line in question. A default value of GFAC=3. is assumed by the program.

The program is designed to act like many filters which sample the data at many points. /FILTER/ controls the nature of the filters. NSAMP, the first element

of /FILTER/ equals the number of bandpasses to be used by the program ($\leq$ 15 and odd).XBAND is the bandpass of each filter in microns wavelength and XSTEP is the distance between bandpasses in microns wavelength. Default values are NSAMP=15, XBAND=.01E-2, XSTEP=.03E-2.

It is possible in this program to envoke an alternate routine for determining field strength. /FFT/ contains the control data for this routine. NOFFT=.TRUE. causes this technique to be skipped altogehter. BMIN sets the lower limit of fields that this routine will consider. GBND gives the number of degrees around $\delta = 90^\circ$ in which this routine is not called. The default values are presently set at NOFFT=.TRUE., BMIN=1800. (gauss), GBNO = 20.(degrees).

4. Algorithms

The details of operation of PASS2, PASS3, and PASS4 are described here. PASS2 and PASS4 are briefly and basically described. PASS 3 is described in greater rigor since this is where the real field determination goes on.

4.1 PASS2

PASS2 has the function of converting microdensitometer raw data into manageable data records. The original microdensitometer data consists of records that represent DENSITOMETER SCAN lines. The final output tape saves only the useful data in records, one slit crossing to a record.

PASS2 has the goal to keep track of label records, correctly set and use control parameters, and breakdown of the data. In PASS2A.FTN the MAIN program is listed. Initialization and the keeping track of control parameters is done by the first part of the program (up to statement 40). The loop DO 60 contains the data examination part of the program.

In the initialization phase two possible directions are taken. If this is the first time PASS2 is being run in a complete run (IFIL1=1) then the program requests initial parameters which it stores in a data file SPARM on disk. If this is not a first run (IFIL1/1) then PASS2 sets the initial parameters by checking the file SPARM already on disk.

The data reduction phase of loop DO 60 first opens data files for reading, then reads sequentially all the records on each of these files. As each record it reads it is processed according to what its code word ICODE implies. The actual breakdown of line data records is called if ICODE is negative. Under this condition the loop DO 54 reads the data from the raw data file in a random access technique (using the subroutine READ). PRCSS then handles each line as it is given to it.

PRCSS goes through a four step operation to break down a data line. First if the line was scanned backwards in a raster scan - the line is flipped (subroutine FLIP). Second the edge of the first slit crossing on the line is found by CR3DEG. Third, the line is stepped through and broken down according to the distance given in the common /FILM/ and results are written on the scratch file

SCRTCH.TMP. Fourth, after information for the two orthogonal polarizations is present in SCRTCH.TMP (takes two line process for new data and a one line process for old data) it is rewritten on tape in a finalized form.

This process continues until all the raw data files on disk are read. PASS2 then closes all its files and deletes SCRTCH.TMP.

4.2 PASS3

The heart of the operation is here. PASS3 takes the data tape created by PASS2 and yields magnetic field strengths, angles, and fluid velocities. In this section the mechanics of the operation is discussed. The theory behind the operation is described in the Fourier Transform method by A. Title and T. Tarbell.

Because of the severe space limitation on disk PASS3 must operate as a two step program. First data is read off of tape. As each data record is read all the essentials of that record (field, angles, velocity) are derived in the routine PRCSS using the routine PRMFFT. As results come out they are recorded in an extremely condensed matter on a large continguous scratch file on disk. Finally after all the raw data has been looked at the computer requests a new tape be mounted for output of the final manetogram. PASS3 then begins its second phase of operation and moves data from the SCRATCH.TMP file onto tape. Before exit SCRATCH.TMP is deleted.

It is vital to understand the structure of the in between scratch file to understand the program mechanics. Because it is contiguous this scratch file can be treated as a large array. IO1 is a routine which gives SCRTCH.TMP this appearance. IO1 references V, B, δ, ϕ according to the coordinate IX, IY. Thus any velocity may be easily read or written for any point in the picture at any time in the execution. As field values are found from the data they are averaged with previous results (if there are any and weighted properly), and returned to SCRTCH.TMP over the old location. During a COMPLETE RUN as many as four values for the field may be determined for each point in the picture. These have been appropriately handled so as to take only one word on disk during the COMPLETE RUN.

The work of PASS3 is carried out in PRMFFT and the routines it calls CONTNA, COMBINE, POLYAN, and BGOOD. In addition there are a set of operational routines that perform mathematical operation for PRMFFT, these are DSUM, COSTRA, ZER, PLINT.

Basically PRMFFT manages the fourier transform method of obtaining $|B|$, cosine γ, and the line of sight velocity. Subroutine CONTRA determines the continuum values. COMBINE determines the line center position, and hence the line of sight velocity; while POLYAN manages the determination of the zero crossing, and cosine γ. The value of the zero crossing is determined by ZER and COSTR calculates the required fourier transforms. The polynomial integration procedure for cosine gamma is performed in PLINT. Finally the values determined by the above routines are reviewed in BGOOD. The routine DSUM adds sets of values for a number of averaging processes.

4.3 PASS4

There are no tricks in PASS4. The algorithms are obvious and straightforward. Given magnetogram data PASS4 can be used to construct magnetogram picutres in any of twelve formats.

MAIN (PASS4A) serves the function of determining which output feature is desired by the user. MAIN only sets certain variables which are then passed to the appropriate routines for handling.

PICTUR creates a file for the densitometer which can later be interpreted as a photograph. It uses the two routines LABEL, and SETSCL to write the label and parameter block of the output file. LNINT is where the work goes on, as this routine determines the actual intensity for the picture point being written. PICTUR may be called when MODE 1, 2, or 3 of the routine are entered.

MODES 4 and 5 of the program cause the routine VECT. to be called from MAIN. VECT examines the magnetogram data and constructs on the output screen a vector magnetogram according to specifications.

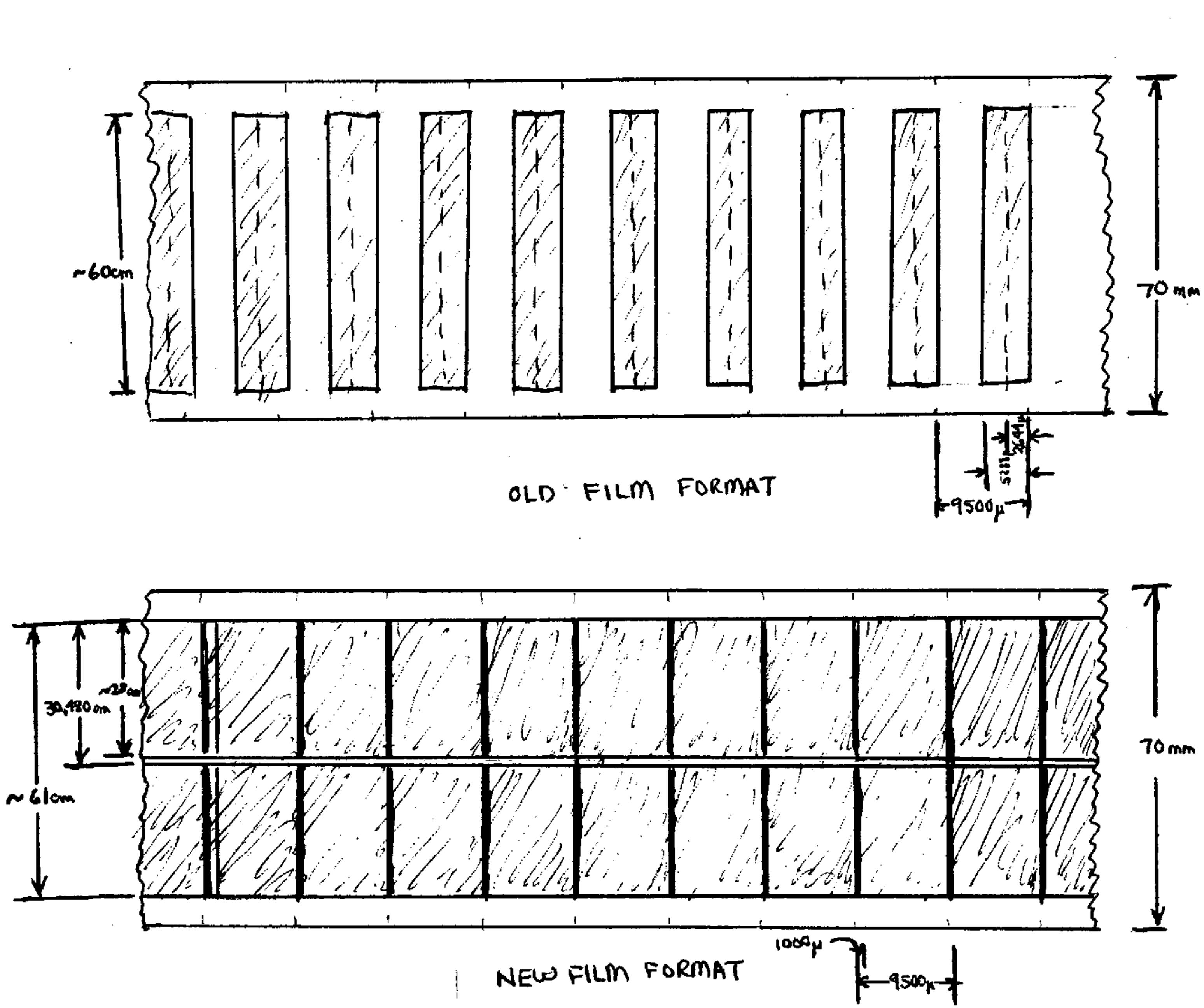
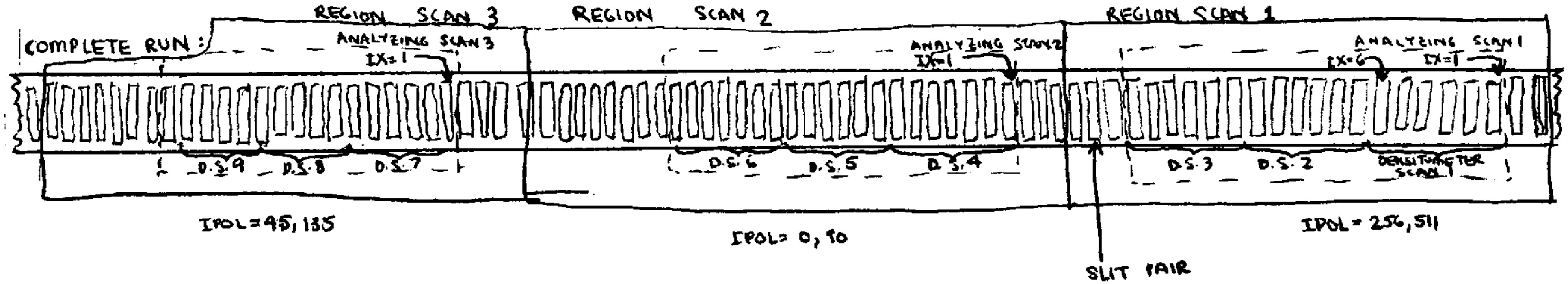


Figure 1.1 Old and New Film Formats for Spectra-Spectraheliograms.



This is an example of a piece of spectra-spectraheliograph film with the terms identified. In the complete run of this example the region was scanned three times in three polarization pairs. REGION SCAN 1 was LHC, RHC (IPOL = 256,511), REGION SCAN 2 was x,y (IPOL = 0,90), REGION SCAN 3 was in linear polarizations $45^\circ, 135^\circ$ (IPOL = 45,135). Corresponding to the three REGION SCANS are three ANALYZING SCANS each of which begin at the same point on the sun. This solar starting point is labeled IX = 1. In this example each ANALYZING SCAN took three separate DENSITOMETER SCANS. Each DENSITOMETER SCAN covered about 7 slit pairs (not clear in drawing, but may vary even in real run).

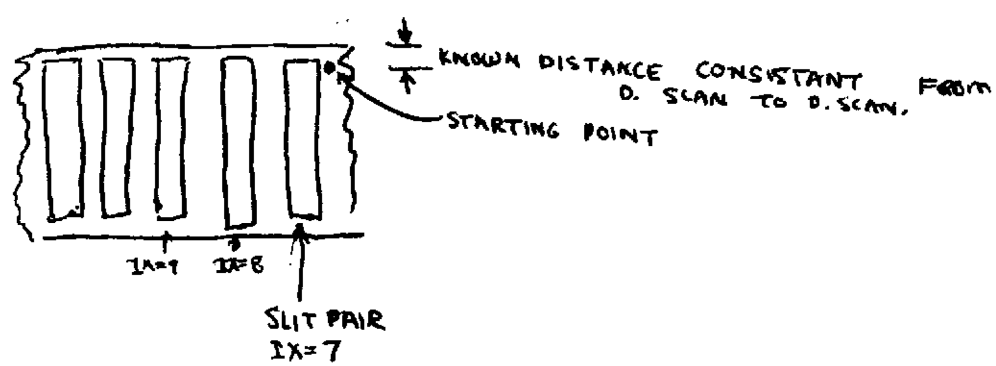


Figure 1.2

```
C     PASS 2 -DATA FROM DISK TO TAPE
C     THIS PROGRAM TAKES A SEQUENCE OF M INPUT FILES LABELED <CHAR>1.DAT
C     TO <CHAR>M.DAT OF DENSITIZED DATA AND PRODUCES AN OUTPUT TAPE
C     OF IDENTICALLY LABELED FILES WITH THE DATA ORGANIZED INTO SMALL
C     RECORDS EACH CONTAINING ONE SLIT CROSSING.  THE OUTPUT DATA TAPE
C     CONTAINS SEQUENCIAL FILES EACH FILE CONTAINING MANY RECORDS.
C     EACH RECORD IS IDENTIFIED BY ITS FIRST WORD =ICODE AND BY ITS
C     SECOND WORD N WHICH TELLS THE LENGTH OF THE BUFFER TO
C     FOLLOW.  WITHIN THE BUFFER THE INFORMATION IS CODED AS FOLLOWS:
C     ICODE=1,N,IBUF=LABEL OF LENGTH N WORDS.
C     ICODE=2,N,IBUF(1 AND 2)=XSTEP=SECONDS OF ARC PER X STEP(REAL).
C      IBUF(3 AND 4)=YSTEP=SECONDS OF ARC PER Y STEP.
C      IBUF(5 AND 6)=ZSTEP=MICRONS OF WAVELENGTH PER DATA POINT.
C      IBUF(7 AND 8)=ZCEN=MICRONS WAVELENGTH OF CENTERLINE.
C      IBUF(9)=NLPF=NUMBER OF YSTEPS IN Y DIMENSION.
C      IBUF(10)=1 IF OLD DATA, 2 IF NEW DATA.
C     ICODE=0,N,IBUF(1)=J=NUMBER OF POINTS IN DATA RECORD=N-6.
C      IBUF(2)=IX=NUMBER OF X STEP.
C      IBUF(3)=IY=NUMBER OF Y STEP.
C      IBUF(4 AND 5)=Z0=NUMBER OF Z STEPS (REAL) FROM FIRST RECORD TO LINE
C       CENTER.
C      IBUF(6)=IPOL=POLARIZATION OF RECORD (511-RHC,256-LHC,0 TO 180=
C       LINEAR).
C      IBUF(7) TO IBUF(N)=FILM DENSITIES ,SMOOTHED AND PHOTOMETERED.
C     RECORDS ARE ORDERED IN ORTHOGONAL POLARITY PAIRS SEQUENTIALLY
C     IN IX FOR K STEPS, FOR EACH VALUE OF IY.
C     LARRY NOVEMBER - 21-AUG-73
        COMMON /BUFFER/ICODE,IBUF(1)
        DIMENSION ZLNE(5),IPOL(2),NUM(8),IEXT(5)
        LOGICAL END
        LOGICAL*1 IFIL(10),IFILNM(4),CHAR(2)
        EQUIVALENCE (IFIL(1),IEXT(1)),(IFIL(3),IFILNM(1)),
       +   (CHAR(1),ICHAR)
        DATA IEXT/'00','00','00','.D','AT'/,ICHAR/' 0'/
        CALL SETFIL(2,'SCRTCH.TMP',IERR,'DK',0,0,"233,2)
        DEFINE FILE 2(50,512,U,IVAR)
        WRITE(2'1,ERR=3) IDUMMY
        GO TO 4
      3 CONTINUE
        END FILE 2
        WRITE(6,1008) IERR
        STOP 1
      4 CONTINUE
        WRITE(6,1000)
        READ(6,902) IFIL(1),IFIL1,IFIL2
        IF(IFIL1.EQ.1) GO TO 6
        CALL SETFIL(1,'SPARM')
        READ(1) NLNE,ZUNIT,XSTEP,YUNIT,ZLNE
        END FILE 1
        GO TO 12
      6 CONTINUE
        WRITE(6,1001)
        READ(6,900) ZUNIT,XSTEP,YUNIT
        WRITE(6,1002)
        READ(6,900) ZLNE(1),ZLNE(2)
```

```
      NLNE=0
      DO 10 L1=3,5
      WRITE(6,1003)
      READ(6,900) ZLNE(L1)
      IF(ZLNE(L1).EQ.0.) GO TO 11
      NLNE=NLNE+1
   10 CONTINUE
   11 CALL SETFIL(1,'SPARM')
      WRITE(1) NLNE,ZUNIT,XSTEP,YUNIT,ZLNE
      END FILE 1
   12 CONTINUE
      DO 60 L0=IFIL1,IFIL2
      ENCODE(4,1006,IFILNM) L0
      DO 13 L1=1,4
   13 IF(IFILNM(L1).EQ.CHAR(1)) IFILNM(L1)=CHAR(2)
      WRITE(6,1007) IFIL
      CALL SETFIL(1,IFIL)
      CALL SETFIL(8,IFIL)
      WRITE(6,1005)
      READ(6,901) IX
      WRITE(6,1004)
      READ(6,901) IPOL
        CALL CURVE
      IREC1=0
      IFR=0
   15 CONTINUE
      IREC1=IREC1+1
      CALL READ(IREC1,N,ICODE,IBUF,END)
      IF(END) GO TO 55
      IF(ICODE.LE.0) GO TO 51
      GO TO (30,30,50,30),ICODE
   30 ICODE=1
      WRITE(8) ICODE,N,(IBUF(L1),L1=1,N)
      GO TO 15
   50 CALL SPARM(IBUF,ZUNIT,XSTEP,YUNIT,NPPL,NLPF,NFR,
     *    LPAT,ZLNE(2),ZSTEP)
      GO TO 15
   51 DO 54 L1=1,NLPF
      DO 54 L2=1,NFR
      CALL PRCSS(ICODE,IBUF,IPOL,IX,L1,ZSTEP,NLNE,ZLNE,
     *    LPAT,NPPL,L2,NFR)
      CALL READ((NFR-L2)*NLPF+L1+IREC1,N,ICODE,IBUF,END)
      IF(END) GO TO 55
   54 CONTINUE
   55 CONTINUE
      END FILE 1
      END FILE 8
   60 CONTINUE
      END FILE 2
      CALL DELETE
  900 FORMAT(E10.0)
  901 FORMAT(I3)
  902 FORMAT(A1,/,I3,/,I3)
 1000 FORMAT(' PLEASE SPECIFY DISK INPUT FILES WITH ONE '/
     *    ' ID CHARACTER <CR> (A1 FORMAT)''/
     *    ' THEN SPECIFY THE FIRST FILE NUMBER IN THE SEQ-'/
```

```
                *    ' UENCE <CR>, AND THE LAST FILE NUMBER  <CR> (I3 FORMATS).'/)
ØØ8Ø      1ØØ1 FORMAT(' INPUT WAVELENGTH SCALE (MICRONS/MICRON ON FILM),'/
                *    ' THEN <CR>, XSTEP (SECONDS OF ARC) BETWEEN SPECTRA,'/
                *    ' THEN <CR>, YUNIT (SECONDS OF ARC PER'/
                *    ' MICRON ON FILM' <CR>.(E1Ø.Ø FORMATS),/)
ØØ81      1ØØ2 FORMAT(' INPUT WAVELENGTH OF FILM LEADING EDGE (MICRONS) <CR>''/
                *    ' WAVELENGTH OF LINE CENTER <CR>.'/)
ØØ82      1ØØ3 FORMAT(' INPUT WAVELENGTHS OF ADDITIONAL LINES ON FILM'/
                *    ' THEN <CR>, OR <CR> IF NO MORE.'/)
ØØ83      1ØØ4 FORMAT(' INPUT POLARIZATION (RHC-511,LHC-256,LINEAR-ANGLE'/
                *    ' IN DEREES FROM RA Ø-18Ø) OF FIRST IN PAIR <CR>,'/
                *    ' OF SECOND IN PAIR <CR> (I3 FORMAT).'/)
ØØ84      1ØØ5 FORMAT(' INPUT IX (SLIT STEPS) FOR FIRST INPUT RECORD'/
                *    ' FROM BEGINNING OF THIS SCAN (>Ø WITH FIRST SLIT CROSSING'/
                *    ' OF REGION SCAN = 1)(I3 FORMAT)'/)
ØØ85      1ØØ6 FORMAT(I4)
ØØ86      1ØØ7 FORMAT(' OPENING FILE ',1ØA1)
ØØ87      1ØØ8 FORMAT(' ERR:FORTØØ1Ø'',I2,'   EXECUTION TERMINATED.')
ØØ88           END

      ROUTINES CALLED:
      SETFIL, CURVE , READ   , SPARM , PRCSS , DELETE

      SWITCHES = /ON

      BLOCK         LENGTH
      MAIN.     1394    (ØØ5344)*
      BUFFER    2       (ØØØØØ4)

      **COMPILER ----- CORE**
         PHASE         USED  FREE
      DECLARATIVES  ØØ366 13425
      EXECUTABLES   ØØ967 12824
      ASSEMBLY      Ø1729 14979
```

```
            SUBROUTINE READ(NEWREC,N,ICODE,IBUF,END)
      C    READ IS A RANDOM ACCESS READ STATEMENT FOR DENSITOMETER
      C    DATA RECORDS.  NEWREC SPECIFIES THE RECORD TO BE READ, AND
      C    ICODE,N,IBUF(1) TO IBUF(N) ARE THE INFORMATION ON THAT RECORD. END
      C    IS SET .TRUE. IF AN ATTEMPT HAS BEEN MADE TO READ BEYOND
      C    THE LAST RECORD OF THE FILE.
            DIMENSION IBUF(1)
            LOGICAL END
            DATA IREC/0/
            END=.FALSE.
            IF(NEWREC-IREC) 30,25,10
         10 IREC=IREC+1
            DO 20 L1=IREC,NEWREC
            READ(1,END=60) ICODE,N,(IBUF(L2),L2=1,N)
         20 CONTINUE
            IREC=NEWREC
         25 RETURN
         30 REWIND 1
            IREC=0
            GO TO 10
         60 END=.TRUE.
            RETURN
            END

      SWITCHES = /ON

      BLOCK        LENGTH
      READ     140     (000430)*

      **COMPILER ----- CORE**
         PHASE         USED  FREE
      DECLARATIVES  00366 13425
      EXECUTABLES   00527 13264
      ASSEMBLY      00996 15712
```

```
      SUBROUTINE SPARM(IBUF,ZUNIT,XSTEP,YUNIT,NPPL,NLPF,NFR,
     *   LPAT,ZCEN,ZSTEP)
C   SPARM EXAMINES THE PARAMETER RECORD OF THE INPUT DENSITOMETER
C   DATA AND RETURNS CERTAIN PARAMETERS ESSENTIAL FOR THE PROGRAM.
C   THESE PARAMETERS ARE:
C    NPPL=NUMBER OF POINTS PER LINE.
C    NLPF=NUMBER OF LINES PER FRAME=NUMBER OF YSTEPS IN SCAN.
C    LPAT=PATTERN=.TRUE. IF RASTER,.FALSE. IF EDGE.
C    NFR=NUMBER OF FRAMES=1 IF OLD DATA, 2 IF NEW DATA.
C   IN ADDITION SPARM SETS /FILM/ COMMON AND WRITES THE PARAMETER RECO
C   FOR THE OUTPUT TAPE.
      COMMON /FILM/SLTSTP(2),SMRSTP(2),SEFF,SLT(2),SMARG(2),ZEFF
      DIMENSION IBUF(1)
      DATA ICODE/2/,N/10/
      NPPL=IBUF(1)
      IDELX=IBUF(4)
      IDELY=IBUF(5)
      NLPF=IBUF(8)
      LPAT=IBUF(9)
      NFR=IBUF(12)/2+1
      IF(NFR.LT.1) NFR=1
      IF(NFR.GT.2) NFR=2
      DELX=IDELX
      DO 10 L1=1,2
      SLTSTP(L1)=SLT(L1)/DELX
      SMRSTP(L1)=SMARG(L1)/DELX
   10 CONTINUE
      YSTEP=YUNIT*FLOAT(IDELY)
      ZSTEP=ZUNIT*DELX
      SEFF=ZEFF/ZSTEP
      WRITE(8) ICODE,N,XSTEP,YSTEP,ZSTEP,ZCEN,NLPF,NFR
      RETURN
      END

ROUTINES CALLED:
FLOAT

SWITCHES = /ON

BLOCK       LENGTH
SPARM   278    (001054)*
FILM    20     (000050)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES 00366 13425
EXECUTABLES  00701 13090
ASSEMBLY     01141 15567
```

```
      SUBROUTINE PRCSS(INIT,IDAT,IPOL,IXØ,IY,ZSTEP,NLNE,ZLNE,
     *   LPAT,NPPL,IFR,NFR)
C   PRCSS EXAMINES DENSITOMETER DATA RECORDS AND PRODUCES
C   SEPERATED OUTPUT DATA RECORDS EACH CONTAINING INFORMATION
C   FOR A SINGLE SLIT CROSSING.
C   THIS IS A FOUR STEP PROCESS.  FIRST, IF THE DATA IS A RASTER BACK
C   SCAN LINE THE DATA IS FLIPPED.  SECOND, CR3DEG FINDS THE
C   EDGE OF THE FIRST SLIT CROSSING ON THE RECORD.  THIRD, THE PROGRAM
C   WALKS THROUGH THE LINE SEPERATING OUT SLIT CROSSINGS AND
C   WRITES THESE ON A SCRATCH FILE.  FOURTH, DATA IS WRITTEN ON TAPE
C   IN A CONVENIENT ORDER FOR LATER USE.
      DIMENSION IDAT(1),IPOL(1),ZLNE(1),ILNE(3),INTG(2)
      LOGICAL LPAT,LDIR,SKIP
      COMMON /FILM/SLT(2),SMARG(2),SEFF
      EQUIVALENCE (INTG(1),REALX)
      DATA ICODE/Ø/
      IF(INIT.EQ.-1) LDIR=.TRUE.
      IF(.NOT.LDIR) CALL FLIP(IDAT,NPPL)
      IF(INIT.EQ.-1) CALL CR3DEG(IDAT,ZERO,NPPL,NFR)
      BUF=ZERO-SMARG(IFR)
      L1=Ø
   2Ø L1=L1+1
      IF(NFR.NE.2) L2=L1-2*((L1-1)/2)
      IF(NFR.EQ.2) L2=IFR
      BUF=BUF+SMARG(L2)
      NBUF=BUF+.5+SLT(L2)
      IF(NBUF.GT.NPPL) GO TO 4Ø
      IREC2=NFR*(L1-1)+IFR
      IDAT(1)=SLT(L2)
      IF(SLT(L2).GT.SEFF) IDAT(1)=SEFF
      IDAT(2)=IXØ+(L1-1)/2
      IDAT(3)=IY
      I1=SLT(L2)-SEFF
      IF(I1.LT.Ø) I1=Ø
      J1=BUF+1.
      J2=BUF+SLT(L2)
      REALX=ZER(IDAT,J1,J2,NLNE,ZLNE,ZSTEP,INIT)-FLOAT(I1/2)
      IDAT(4)=INTG(1)
      IDAT(5)=INTG(2)
      IDAT(6)=IPOL(L2)
      CALL PHOT(IDAT,J1,J2)
      I1=SLT(L2)
      L4=BUF
        DBUF=BUF-FLOAT(L4)
        IWAY=+1
        IF(DBUF.GE.Ø.) GO TO 23
        IWAY=-1
        DBUF=ABS(DBUF)
   23   DBUFI=1.-DBUF
      IF(SLT(L2).LE.SEFF) GO TO 22
      I1=SEFF
      L4=BUF+SEFF/2.
   22 J1=7
      J2=I1+6
      DO 3Ø L3=J1,J2
```

```
            L4=L4+1
         30 IDAT(L3)=DBUF*IDAT(L4+IWAY)+DBUFI*IDAT(L4)
            BUF=BUF+SLT(L2)
            WRITE(2'IREC2) J2,(IDAT(L3),L3=1,J2)
            GO TO 20
         40 CONTINUE
            IF(IFR.LT.NFR) RETURN
            IF(IREC2-2*(IREC2/2).NE.0) IREC2=IREC2-1
            DO 50 L1=1,IREC2
            READ(2'L1) I1,(IDAT(L2),L2=1,I1)
            WRITE(8) ICODE,I1,(IDAT(L2),L2=1,I1)
         50 CONTINUE
            IF(LPAT) LDIR=.NOT.LDIR
            RETURN
            END

      ROUTINES CALLED:
      FLIP   , CR3DEG, ZER    , FLOAT , PHOT   , ABS

      SWITCHES = /ON

      BLOCK        LENGTH
      PRCSS    855     (003256)*
      FILM     10      (000024)

      **COMPILER --*-- CORE**
         PHASE        USED  FREE
      DECLARATIVES 00366 13425
      EXECUTABLES  00948 12843
      ASSEMBLY     01513 15195
```

```
ØØØ1          SUBROUTINE FLIP(IDAT,N)
        C   FLIP TAKES AN ARRAY IDAT OF LENGTH N AND REVERSES THE ORDER OF
        C   THE ARRAY ELEMENTS.
ØØØ2          DIMENSION IDAT(1)
ØØØ3          NP1=N+1
ØØØ4          MID=N/2
ØØØ5          DO 1Ø L1=1,MID
ØØØ6          L1P=NP1-L1
ØØØ7          IHLD=IDAT(L1P)
ØØØ8          IDAT(L1P)=IDAT(L1)
ØØØ9          IDAT(L1)=IHLD
ØØ1Ø       1Ø CONTINUE
ØØ11          RETURN
ØØ12          END

        SWITCHES = /ON

        BLOCK         LENGTH
        FLIP     1Ø7     (ØØØ326)*

        **COMPILER ----- CORE**
           PHASE         USED  FREE
        DECLARATIVES  ØØ366 13425
        EXECUTABLES   ØØ527 13264
        ASSEMBLY      ØØ94Ø 15768
```

```
      FUNCTION ZER(IDAT,J1,J2,NLNE,ZLNE,ZSTEP,INIT)
C    ZER DETERMINES THE DISTANCE FROM THE BEGINNING OF THE SLIT
C    CROSSING TO LINE CENTER BY CALIBRATING OFF OF LINE EDGE AND ANY
C    OTHER REFERENCE LINES IN THE FRAME.
      DIMENSION IDAT(1),ZLNE(1)
      LOGICAL INIT
      IF(NLNE.NE.0) GO TO 10
      IF(.NOT.INIT) RETURN
      ZER=(ZLNE(2)-ZLNE(1))/ZSTEP
      RETURN
   10 CONTINUE
      ZC=1.
      DO 40 L1=1,NLNE
      ILNE=(ZLNE(L1+2)-ZLNE(1))/ZUNIT-10.
      J2P=ILNE+50
      J2M=ILNE+10
      IDIFM=0
      DO 20 L2=ILNE,J2P
      IDIF=IDAT(L2+J1+1)-IDAT(L2+J1)
      IF(IDIF.LT.IDIFM) GO TO 20
      IDIFM=IDIF
      L2M=L2
   20 CONTINUE
   40 ZC=ZC+FLOAT(L2M)*ZSTEP/(ZLNE(L1+2)-ZLNE(1))
      ZER=ZC/FLOAT(NLNE+1)*(ZLNE(2)-ZLNE(1))/ZSTEP
      RETURN
      END

ROUTINES CALLED:
FLOAT

SWITCHES = /ON

BLOCK        LENGTH
ZER     326     (001214)*

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES 00446 13345
EXECUTABLES  00607 13184
ASSEMBLY     01132 15576
```

```
FORTRAN V004A                          00:26:01     01-JAN-72    PAGE    1

              SUBROUTINE CR3DEG(IDAT,ZER,NPPL,NFR)
      C    CR3DEG FINDS THE EDGE OF THE FIRST SLIT CROSSING IN THE DATA ARRAY
      C    IDAT.  IT LOOKS FOR A SLOPE IN FILM DENSITIES THAT IS GREATER
      C    THAN ICONTR OVER TEN POINTS.
              DIMENSION IDAT(1)
              COMMON /DENS/ICONTR
              IDIFF(N,M)=(IDAT(N)-IDAT(N+M))*ISGN
              ISGN=1
              IF(NFR.EQ.2) ISGN=-1
              LIMITN=NPPL/2
              DO 20 L1=1,LIMITN
              IDIF=IDIFF(L1,10)
              IF(IDIF.GT.ICONTR) GO TO 30
           20 CONTINUE
              WRITE(6,1000)
              STOP 2
           30 NZER=L1-1
              IDIFM=0
              L1M=0
              DO 40 L1=1,10
              IDIF=IDIFF(NZER+L1,1)
              IF(IDIF.LT.IDIFM) GO TO 40
              IDIFM=IDIF
              L1M=L1
           40 CONTINUE
              NZER=NZER+L1M
              ZER=NZER
              RETURN
         1000 FORMAT(' ERR:CR3DEDG COULD NOT FIND EDGE. EXECUTION TERMINATED')
              END

      SWITCHES = /ON

      BLOCK        LENGTH
      CR3DEG  286     (001074)*
      DENS      1     (000002)

      **COMPILER ----- CORE**
         PHASE       USED  FREE
      DECLARATIVES  00366 13425
      EXECUTABLES   00616 13175
      ASSEMBLY      01245 15463
```

```
                SUBROUTINE CURVE
      C         PHOTOMETRY  FINDS 1024*EXPOSURE AS FCN OF TRANS./2
      C         USES HARVARD PHOTOMETRY CURVE OF JUNE, 1971 (5 PERCENT)
      C         T. TARBELL  NOV. 19, 1973
                COMMON/CCURV/IPHOT(513),KSMOOT
                DATA C1,C2,C3,A0,A1,A2,SLOPE,EINT,B0,B1,B2,S2,E2/
              1 2.017,0.408,0.1115,101.61,218.26,-13.1,161.63,162.53,
              2 44.46,-7740.5,69480.0,1571.07,-123.87/
                DO 10 I=1,513
                F=FLOAT(I)
                F=ALOG(500./F)+0.1
                IF (F.GT.C1) GO TO 1
                IF (F.GT.C2) GO TO 2
                IF (F.GT.C3) GO TO 3
                IPHOT(I) = S2*F + E2
                IF(IPHOT(I).LT.1) IPHOT(I)=1
                GO TO 9
      1         IPHOT(I) = SLOPE*F + EINT
                IF(IPHOT(I).GT.1023) IPHOT(I) = 1023
                GO TO 9
      2         IPHOT(I) = A0 +A1*F + A2*F*F
                GO TO 9
      3         IPHOT(I) = B0 + SQRT(B1 + B2*F)
      9         IPHOT(I) = 1.40*IPHOT(I)
                IPHOT(I) = MAX0(1,1024-IPHOT(I))
                IPHOT(I)=MIN0(1023,IPHOT(I))
      10        CONTINUE
                WRITE (6,100)
      100       FORMAT(/' TYPE 1 FOR SMOOTHING, 0 FOR NONE'/)
                READ (6,101) KSMOOT
      101       FORMAT (I2)
                RETURN
                END

      ROUTINES CALLED:
      FLOAT , ALOG  , SQRT   , MAX0   , MIN0

      SWITCHES = /ON

      BLOCK      LENGTH
      CURVE   422    (001514)*
      CCURV   514    (002004)

      **COMPILER ----- CORE**
         PHASE       USED  FREE
      DECLARATIVES 00537 13254
      EXECUTABLES  00687 13104
      ASSEMBLY     01397 15311
```

```
FORTRAN V004A                          00:26:38     01-JAN-72    PAGE    1

0001              SUBROUTINE PHOT(IDAT,J1,J2)
          C       CONVERTS FROM TRANSMISSION TO (1024*EXPOSURE)
          C       T. TARRELL  NOV. 19, 1973
0002              DIMENSION IDAT(1), RAT(10)
0003              COMMON /WEDGE/NSTEPS,NOWEG,ITRANS(11),INT(11)
0004              COMMON /CCURV/IPHOT(513),KSMOOT
0005              LOGICAL NOWEG
0006              DO 100 L=J1,J2
0007              LL=IDAT(L)/2
0008              IF(LL.LE.0) LL=1
0009              IF(LL.GE.513) LL=513
0010      100     IDAT(L)=IPHOT(LL)
0011              IF(KSMOOT.EQ.0) RETURN
0012              M2=J1+2
0013              M3=J2-2
0014              DO 200 L1=M2,M3
0015      200     IDAT(L1)=0.4*(IDAT(L1)+0.5*(IDAT(L1+1)+IDAT(L1-1)+
               1  0.5*(IDAT(L1+2)+IDAT(L1-2))))
0016              RETURN
0017              END

        SWITCHES = /ON

        BLOCK        LENGTH
        PHOT    241     (000742)*
        WEDGE   24      (000060)
        CCURV   514     (002004)

        **COMPILER ----- CORE**
           PHASE        USED  FREE
        DECLARATIVES  00549 13242
        EXECUTABLES   00607 13184
        ASSEMBLY      01098 15610
```

```
0001          BLOCK DATA
      C     ALL THE PROGRAM CONTROL PARAMETERS ARE SET HERE.  THESE PARA-
      C     ARE DESCRIBED:
      C      /FILM/=CONTAINS INFORMATION ON FILM DIMENSIONS.
      C       SLT(1 AND 2)=MICRONS ACROSS SLIT FOR FIRST (1) AND SECOND (2)
      C       ORTHOGONAL POLAIZATIONS.
      C       SMARG(1 AND 2)=MICRONS IN MARGIN PRIOR TO SLIT CROSSING.
      C       ZEFF=NUMBER OF MICRONS OF USEFUL DATA AROUND LINE CENTER.
      C      /DENS/=ICONTR=MINIMUM CONTRAST BETWEEN SLIT EDGE AND MARGIN.
      C      /WEDGE/=STEP WEDGE DATA.
      C       NOWEG=.TRUE. IF NO WEDGE INFORMATION IS AVAILABLE
      C       NSTEPS=NUMBER OF WEDGE STEPS.
      C       ITRANS(K)=TRANSMISSION OF K-TH STEP IN WEDGE.
      C       INT(K)=CORRECT TRANSMISSON OF K-TH STEP IN WEDGE.
0002          COMMON /FILM/DUM1(5),SLT(2),SMARG(2),ZEFF
             *   /DENS/ICONTR
             *   /BUFFER/IC,IBUF(10240)
             *   /WEDGE/NSTEPS,NOWEG,ITRANS(11),INT(11)
0003          LOGICAL NOWEG
0004          DATA ICONTR/200/
0005          DATA SMARG,SLT/3527.,685.,2644.,2644./,ZEFF/0.8/
0006          DATA NSTEPS/11/,ITRANS/0,100,200,300,400,500,600,
             *   700,800,900,1000/,INT/0,100,200,300,400,500,600,700,
             +   800,900,1000/
0007          DATA NOWEG/.TRUE./
0008          END

      SWITCHES = /ON

      BLOCK        LENGTH
      DATA.    0        (000000)*
      FILM     20       (000050)
      DENS     1        (000002)
      BUFFER   10241    (050002)
      WEDGE    24       (000060)

      **COMPILER --*-- CORE**
         PHASE         USED  FREE
      DECLARATIVES  00366 13426
      EXECUTABLES   00570 13221
      ASSEMBLY      00860 15848
```

```
C     PASS 3 DATA FROM TAPE TO FINAL TAPE FORM
C     THIS PROGRAM TAKES A SEQUENCE OF INPUT FILES LABELED <CHAR>1.DA
C     THROUGH <CHAR>M.DAT OFF OF TAPE WHICH HAS BEEN MADE BY PASS2 AND
C     DETERMINES THE MAGNETIC FIELDS B, FIELD ANGLES (GAM,PHI), AND
C     VELOCITIES V FOR ALL THE POINTS IN THE REGION.  PASS3 STORES THE
C     RESULTS OF ITS ANALYSIS IN A SCRATCH FILE UNTIL COMPLETION OF
C     THE ANALYSIS.  AT COMPLETION OF THE ANALYSIS PASS3 REQUESTS AN OUT
C     TAPE AND AFTER CONTINUATION PUTS THE DATA ON TAPE.  THE FINAL TAPE
C     RECORDS BEGIN WITH TWO KEY WORDS ICODE AND N.  ICODE
C     DEFINES THE NATURE OF THE RECORD AND N THE LENGTH OF THE BUFFER TO
C     FOLLOW.  THE OUTPUT RECORDS ARE CODED AS FOLLOWS:
C     ICODE=1,N,IBUF=LABEL OF LENGTH N CHARACTERS.
C     ICODE=2,N,IBUF(1&2)=XSTEP=SECONDS OF ARC BETWEEN X STEPS(REAL).
C      IBUF(3&4)=YSTEP=SECONDS OF ARC BETWEEN Y STEPS(REAL).
C      IBUF(5)=NX=NUMBER OF FIELD UNITS IN X DIMENSION.
C      IBUF(6)=NY=NUMBER OF FIELD UNITS IN Y DIMENSION.
C     ICODE=Ø,N,IBUF(1)=IX=X STEP POSITON OF THIS DATA POINT.
C      IBUF(2)=IY=Y STEP POSITION OF THIS DATA POINT.
C      IBUF(3&4)=V=VELOCITY AT THIS POINT(KM/SEC, +=DOWN)(REAL).
C      IBUF(5&6)=B=FIELD STRENGTH AT THIS POINT(GAUSS)(REAL).
C      IBUF(7&8)=GAM=FIELD ANGLE FROM LINE OF SIGHT(DEGREES)(REAL).
C      IBUF(9&1Ø)=PHI=FIELD ANGLE FROM X SCAN DIRECTION(DEGREES)(REAL).
C     LARRY NOVEMBER - 21-AUG-73
        COMMON /BUFFER/IBUFC(512)
        DIMENSION IBUF(256,2),IBUF1(256),IBUF2(256),IEXT(5),LABEL(32)
        LOGICAL*1 CHAR(2),IFIL(1Ø),IFILNM(4)
        EQUIVALENCE (IBUFC(1),IBUF(1,1),IBUF1(1)),(IBUFC(257),IBUF2(1)),
      *    (IFIL(1),IEXT(1)),(IFIL(3),IFILNM(1)),(CHAR(1),ICHAR)
        DATA IEXT/'ØØ','ØØ','ØØ','.D','AT'/,ICHAR/' Ø'/
        CALL SETFIL(2,'SCRTCH.TMP',IERR,'DK',Ø,Ø,"233,2)
        WRITE(6,1ØØØ)
        READ(6,9Ø1) NX,NY
        WRITE(6,1ØØ1)
        READ(6,9ØØ) IFIL(1)
        L1=Ø
      1 L1=L1+1
        ENCODE(4,1ØØ3,IFILNM) L1
        DO 3 L2=1,4
      3 IF(IFILNM(L2).EQ.CHAR(1)) IFILNM(L2)=CHAR(2)
        CALL SETFIL(8,IFIL)
      5 CONTINUE
        READ(8,END=1ØØ,ERR=2ØØ) ICODE,N,(IBUF1(L2),L2=1,N)
        I1=ICODE+1
        GO TO (1Ø,2Ø,4Ø),I1
     1Ø READ(8,END=1ØØ) ICODE,N,(IBUF2(L2),L2=1,N)
        CALL PRCSS(IBUF,NX,ZSTEP)
        GO TO 5
     2Ø IF(L1.NE.1) GO TO 5
        DO 3Ø L2=1,N
     3Ø LABEL(L2)=IBUF1(L2)
        LABEL(32)=N
        GO TO 5
     4Ø IF(L1.NE.1) GO TO 5
        CALL SPARM3(IBUF1,NY,XSTEP,YSTEP,ZSTEP)
        CALL IO1(3,IBUF1,NX,NY,IERR)
```

```
0032        NX=0
0033        GO TO 5
0034    100 END FILE 8
0035        GO TO 1
0036    200 CONTINUE
0037        DO 210 L1=1,10
0038        CALL BELL
0039        DO 210 L2=1,10000
0040    210 CONTINUE
0041        WRITE(6,1002)
0042        PAUSE 1
0043        ICODE=1
0044        N=LABEL(32)
0045        WRITE(8) ICODE,N,(LABEL(L1),L1=1,N)
0046        ICODE=2
0047        N=6
0048        WRITE(8) ICODE,N,XSTEP,YSTEP,NX,NY
0049        ICODE=0
0050        N=10
0051        DO 250 L1=1,NY
0052        DO 250 L2=1,NX
0053        CALL IO1(1,IBUF1,L2,L1,IREC)
0054        V=FLOAT(IBUF1(IREC+1))/256.
0055        B=FLOAT(IBUF1(IREC+2))/4.
0056        GAM=FLOAT(IBUF1(IREC+3))/128.
0057        PHI=FLOAT(IBUF1(IREC+4))/128.
0058    250 WRITE(8) ICODE,N,L2,L1,V,B,GAM,PHI
0059        END FILE 8
0060        END FILE 2
0061        CALL DELETE
0062    900 FORMAT(A1)
0063    901 FORMAT(I3)
0064   1000 FORMAT(' SPECIFY THE MAXIMUM NUMBER OF PICTURE',
          *    ' ELEMENTS IN X <CR>,'/
          *    ' AND THE MAXIMUM NUMBER OF PICTURE',
          *    ' ELEMENTS IN Y <CR>'/' (I3 FORMATS).'/)
0065   1001 FORMAT(' SPECIFY ONE LETTER WHICH IDENTIFIES TAPE',
          *    ' INPUT FILES<CR>,'/)
0066   1002 FORMAT(' MOUNT TAPE FOR OUTPUT OF MAGNETOGRAM AND ASSIGN'/
          *    ' TAPE FILE TO UNIT 8, THEN CONTINUE.'/)
0067   1003 FORMAT(I4)
0068        END

    ROUTINES CALLED:
    SETFIL, PRCSS , SPARM3, IO1    , BELL   , FLOAT , DELETE


    SWITCHES = /ON

    BLOCK        LENGTH
    MAIN.    995    (003706)*
    BUFFER   512    (002000)

    **COMPILER ----- CORE**
       PHASE       USED  FREE
    DECLARATIVES  00366 13425
    EXECUTABLES   00911 12880
    ASSEMBLY      01629 15079
```

```
FORTRAN V004A                              01:47:46    20-MAR-74    PAGE     1

0001          SUBROUTINE SPARM3(BUF,NY,XSTEP,YSTEP,ZSTEP)
        C   SPARM3 INTERPRETS THE PARAMETER RECORD ON THE INPUT
        C   TAPE AND RETURNS TO THE PROGRAM THE FOLLOWING PARAMETERS:
        C    XSTEP=SECONDS OF ARC PER X INCREMENT.
        C    YSTEP=SECONDS OF ARC PER Y INCREMENT.
        C    ZSTEP=MICONS WAVELENGTH BETWEEN DATA POINTS.
        C    NY=TOTAL NUMBER OF STEPS IN Y DIMENSION.
        C    DVEL=KM/SEC PER MICRON WAVELENGTH OF RED SHIFT.
        C    DFLD=GAUSS PER MICRON WAVELENGTH OF COMPONENT SHIFT
0002          DIMENSION BUF(4),INTG(2)
0003          COMMON /LNPRM/DVEL,DFLD,GFAC
0004          EQUIVALENCE (INTG(1),REALX)
0005          XSTEP=BUF(1)
0006          YSTEP=BUF(2)
0007          ZSTEP=BUF(3)
0008          ZCEN=BUF(4)
0009          REALX=BUF(5)
0010          NY=INTG(1)
0011          DVEL=2.998E5/ZCEN
0012          DFLD=2.1420E10/GFAC/ZCEN**2
0013          RETURN
0014          END

      SWITCHES = /ON

      BLOCK         LENGTH
      SPARM3   135     (000416)*
      LNPRM    6       (000014)

      **COMPILER ----- CORE**
         PHASE         USED  FREE
      DECLARATIVES   00366 13425
      EXECUTABLES    00628 13163
      ASSEMBLY       00985 15723
```

```
0001        SUBROUTINE PRCSS(IBUF,NX,ZSTEP)
      C   PRCSS DETERMINES THE FIELD STRENGTH, DIRECTION, AND VELOCITY
      C   FROM THE RAW DATA IBUF USING THE THREE ROUTINES PARM,FIELD, AND
      C   PRMFFT.
0002        DIMENSION PK(2,15),IBUF(256,2),IZ0(2),DAT(256),IPOL(2),
           *   PKH(15),DATA(256,2)
0003        EQUIVALENCE (IZ0(1),Z0)
0004        COMMON /FILTER/NSAMP /FFT/NOFFT,BMIN,GBND
0005        LOGICAL NOFFT
0006        DATA NTIME/-1/,NANG/-1/
0007        DO 50 L1=1,2
0008        N=IBUF(1,L1)
0009        IX=IBUF(2,L1)
0010        IY=IBUF(3,L1)
0011        IZ0(1)=IBUF(4,L1)
0012        IZ0(2)=IBUF(5,L1)
0013        IPOL(L1)=IBUF(6,L1)
0014        DO 10 L2=1,N
0015     10 DAT(L2)=IBUF(L2+6,L1)
0016        DO 30 L2=1,N
0017     30 DATA(L2,L1)=DAT(L2)
0018     50 CONTINUE
0019        IF((IX.EQ.1).AND.(IY.EQ.1)) NTIME=NTIME+1
0020        IF((IX.EQ.1).AND.(IY.EQ.1).AND.(IPOL(1).LT.256)) NANG=NANG+1
0021        IF((NTIME.LT.0).OR.((IPOL(1).LT.256).AND.(NANG.LT.0))) RETURN
0022        IF(IX.GT.NX) NX=IX
0023        CALL IO1(1,IBUF,IX,IY,IREC)
0024        GAM=FLOAT(IBUF(IREC+3,1))/128.
0025          PHI=0.
0026          CALL PRMFFT(DATA,N,ZSTEP,Z0,V,B,GAM,CONIN)
0027          B=INT(B/10.)
0028          B=B*10.+CONIN/100.
0029        I1=V*256.
0030        I2=B*4.
0031        I3=GAM*128.
0032        I4=PHI*128.
0033        IBUF(IREC+1,1)=(NTIME*IBUF(IREC+1,1)+I1)/(NTIME+1)
0034        IF(IPOL(1).GE.256) IBUF(IREC+2,1)=I2
0035     65 IF(IPOL(1).GE.256) IBUF(IREC+3,1)=I3
0036        IF(IPOL(1).LT.256) IBUF(IREC+4,1)=
           *   (NANG*IBUF(IREC+4,1)+I4)/(NANG+1)
0037        CALL IO1(2,IBUF,IX,IY,IREC)
0038        RETURN
0039        END

      ROUTINES CALLED:
      IO1    , FLOAT , PRMFFT, INT

      SWITCHES = /ON

      BLOCK        LENGTH
      PRCSS    2244     (010610)*
      FILTER   1        (000002)
      FFT      5        (000012)

      **COMPILER -----  CORE**
         PHASE       USED  FREE
      DECLARATIVES  00637 13154
      EXECUTABLES   00847 12944
```

```
      SUBROUTINE IO1(MODE,IBUF,IX,IY,IREC)
C   IO SCRATCH IS HANDLED BY IO1.  THREE MODES ARE POSSIBLE, MODE=
C   READS INTO IBUF THE REQUESTED COORDINATE IX,IY DATA.  THAT DAT
C   IS CONTAINED IN THE FOUR RECORDS FOLLOWING THE IREC ELEMENT IN ARR
C   IBUF.  MODE=2 WRITES ON SCRATCH IBUF.  MODE=3 INITIALIZES THE FILE
C   TO ACCEPT IX BY IY ELEMENTS.
      DIMENSION IBUF(1)
      X=IX
      Y=IY
      GO TO (100,200,300),MODE
  100 CONTINUE
      IF((IX.LE.0).OR.(IY.LE.0)) RETURN
      REC=(X-1.)*YDIM+Y
      ISEG=(REC-1.)/64.+1.
      IREC=(REC-FLOAT(ISEG-1)*64.-1.)*4.
      READ(2'ISEG) (IBUF(L1),L1=1,256)
      RETURN
  200 CONTINUE
      IF((IX.LE.0).OR.(IY.LE.0)) RETURN
      REC=(X-1.)*YDIM+Y
      ISEG=(REC-1.)/64.+1.
      WRITE(2'ISEG) (IBUF(L1),L1=1,256)
      RETURN
  300 CONTINUE
      YDIM=Y
      REC=X*Y
      ISEG=(REC-1.)/64.+1.
      DEFINE FILE 2(ISEG,256,U,IVAR)
      WRITE(2'1,ERR=350) IDUMMY
      RETURN
  350 END FILE 2
      CALL DELETE
      WRITE(6,1000) IREC
      STOP 1
 1000 FORMAT(' ERR:FORT00100',I2,' EXECUTION TERMINATED.')
      END

ROUTINES CALLED:
FLOAT , DELETE

SWITCHES = /ON

BLOCK      LENGTH
IO1     383    (001376)*

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES 00366 13425
EXECUTABLES  00607 13184
ASSEMBLY     01300 15408
```

```
      SUBROUTINE PRMFFT(DATA,N,ZSTEP,Z0,V,B,GAMF,CONIN)
C     MASTER CALLING ROUTINE FOR FT ANALYSIS OF ZEEMAN PROFS
C     NEEDS CONCEN,ZER,POLYAN
      DIMENSION DATA(512)
      DIMENSION GOLD(2,9)
      COMMON /LNPRM/DVEL,DFLD,GFAC
      COMMON /RESLTS/X1,COSGAM,GAM,SLOPE,IZERO,NORAT,
     1 PLCOF(5),ACOF,BCOF,CCOF,ALOR
      COMMON /STUFF/CONT(4),NCONT(4),COMT(2),ISTART(4),
     1 IEND(4),SUM(140),PHD,DELNC,SHIFT,NSHIFT,SCHIFF,
     2 NCENT,LCENT,SCALE,P,Q,LIMIT,AREA
      DATA DOPVEL/0./
  107  FORMAT (' N, P, Q  ',I4,2F10.4)
      DO 9 I=1,N
      J=I+256
      DATA(J)=1024.-DATA(J)
      DATA(I)=1024.-DATA(I)
  9   CONTINUE
  101 FORMAT(' ',10F10.0)
      M3=256+N
C     PRINT 101, (DATA(I),I=1,N)
C     PRINT 101, (DATA(I),I=257,M3)
      DO 7 I=1,2
      DO 7 J=1,9
  7   GOLD(I,J)=0.
      V=0.
      LCENT=0
      NSHIFT=256
      NCENT=Z0+0.5+DOPVEL
      DOPVEL=0.
      P=0.05*(DATA(7)-DATA(NCENT))
      Q=1.1*P
      ISTART(1)=Z0/2+1
      ISTART(2)=(N+Z0)/2-1
      ISTART(3)=ISTART(1)+256
      ISTART(4)=ISTART(2)+256
      IEND(1)=6
      IEND(2)=N-5
      IEND(3)=262
      IEND(4)=251+N
      CALL CONTNA(DATA,N)
  5   CONTINUE
      COMT(1)=AMAX1(CONT(1),CONT(2))
      CONIN=COMT(1)
      COMT(2)=AMAX1(CONT(3),CONT(4))
C *****FUDGE TO THROW OUT UNDEREXPOSED FRAMES ****
      DEXP=110.
      IF(ABS(COMT(2)-DATA(NCENT+NSHIFT)).LE.DEXP)
     1 GO TO 1054
      M=0
      DO 21 J=1,257,256
      M=M+1
      M3=J+N-1
      DO 21 L=J,M3
  21  DATA(L)=COMT(M)-DATA(L)
```

```
      DMAX1=0.
      M5=NCONT(1)
      M15=NCONT(2)
      DO 2300 I=M5,M15
      IF(DATA(I).LE.DMAX1) GO TO 2300
      MM1=I
      DMAX1=DATA(I)
 2300 CONTINUE
      DMAX1=DSUM(MM1-2,MM1+2,DATA)
      DMAX2=0.
      M5=NCONT(3)
      M15=NCONT(4)
      DO 2301 I=M5,M15
      IF(DATA(I).LE.DMAX2) GO TO 2301
      MM1=I
      DMAX2=DATA(I)
 2301 CONTINUE
      DMAX2=DSUM(MM1-2,MM1+2,DATA)
      P=ALOG(DMAX2)/ALOG(DMAX1)
      IF(ABS(1.-P).GE.0.10) GO TO 2305
      DO 2302 I=1,256
 2302 DATA(I)=ABS(DATA(I))**P
      GO TO 2307
 2305 PRINT 2306,P
 2306 FORMAT (' P WRONG ',F10.2)
 2307 CONTINUE
      CALL COMBIN(DATA,N)
      DOPVEL=LCENT+PHD-Z0
C   **** FUDGE FACTOR FOR 32 MICRON DATA PTS...5250
      V=0.2528*DOPVEL
      LA=LCENT-NCONT(1)
      LB=IABS(NCONT(2)-LCENT)
      LC=LCENT+NSHIFT-NCONT(3)
      LD=IABS(NCONT(4)-LCENT-NSHIFT)
      LIM=MIN0(LA,LB,69)-1
      DO 10 KMODE=1,2
      LIMIT=LIM-1
      NORIG=LIMIT+1
  104 FORMAT (' ',7I6)
      DO 964 L=1,LIMIT
      SUM(NORIG+L)=DATA(LCENT+L)
  964 SUM(NORIG-L)=DATA(LCENT-L)
      SUM(NORIG)=DATA(LCENT)
      CALL POLYAN(SUM,140,NORIG,LIMIT,PHD,KMODE)
C    ***FUDGE FACTOR FOR 32 MICRON INTERVAL   5250 ***
      GOLD(KMODE,4)=SLOPE*113.92
      IF(IZERO.EQ.0) GO TO 71
      GOLD(KMODE,1)=X1
      GOLD(KMODE,2)=X1*113.92
      IF(NORAT.NE.0) GO TO 71
      GOLD(KMODE,3)=COSGAM
      GOLD(KMODE,5)=COSGAM*GOLD(KMODE,2)
      GOLD(KMODE,6)=PLCOF(2)
      GOLD(KMODE,7)=(ABS(PLCOF(3))+ABS(PLCOF(4))+ABS(PLCOF(5)))
     1 *0.33333/ABS(PLCOF(2))
      IF(ABS(COSGAM).LT.0.7) GO TO 71
```

```
Ø100            CGM=GOLD(KMODE,4)/GOLD(KMODE,2)
Ø101            GOLD(KMODE,3)=Ø.33*COSGAM+Ø.67*CGM
Ø102            GOLD(KMODE,5)=GOLD(KMODE,3)*GOLD(KMODE,2)
Ø103      71    CONTINUE
Ø104            LCENT=LCENT+NSHIFT
Ø105            LIM=MINØ(LC,LD,69)-1
Ø106      1Ø    CONTINUE
Ø107            LCENT=LCENT-2*NSHIFT
          C     PRINT 1ØØ,NCONT,CONT,LCENT,DOPVEL,V,
          C   1 ((GOLD(I,J),J=1,8),I=1,2)
Ø108      1Ø54  CONTINUE
Ø109            CALL BGOOD(GOLD,B,GAMF)
Ø11Ø      1ØØ   FORMAT('Ø',4I5,4F1Ø.1/,  LCENT, DOPVEL, V '
              1 I4,2F1Ø.3/(' '3F1Ø.1,5F1Ø.3)//)
Ø111            RETURN
Ø112            END

          ROUTINES CALLED:
          CONTNA, AMAX1 , ABS    , DSUM   , ALOG   , COMBIN, IABS
          MINØ  , POLYAN, BGOOD

          SWITCHES = /ON

          BLOCK        LENGTH
          PRMFFT  1532   (ØØ577Ø)*
          LNPRM   6      (ØØØØ14)
          RESLTS  28     (ØØØØ7Ø)
          STUFF   324    (ØØ121Ø)

          **COMPILER ----- CORE**
             PHASE        USED  FREE
          DECLARATIVES  ØØ823 12968
          EXECUTABLES   Ø1247 12544
          ASSEMBLY      Ø2Ø67 14641
```

```
0001              SUBROUTINE CONTNA(DATA,NPTS)
            C   FINDS CONTINUA AT NCONT(I),CONT INT = CONT(I)
            C   ISTART(4) CONTAINS INDICES OF FIRST PT. TESTED.  IEND(4), LAST PT.
            C   P IS MAX. ALLOWED EXPOSURE CHANGE AT +-2 PTS.FROM PT. TESTED.
            C   Q, LIKEWISE AT +-4 PTS. AWAY.
            C   MODIFIED 1/25/74  CONT(I) FOUND BY LOOKING FOR FLATNESS
            C   NCONT(I) FROM MAX. SIGNAL NEAR FRAME EDGE
            C   NEEDS FCN. DSUM
            C   T. TARBELL  JULY 12,1971.

0002              COMMON /STUFF/CONT(4),NCONT(4),COMT(2),ISTART(4),
                1 IEND(4),SUM(70),DIT(70),PHD,DELNC,SHIFT,NSHIFT,SCHIFF,
                2 NCENT,LCENT,SCALE,P,Q,LIMIT,AREA
0003              DIMENSION DATA(512)
0004              DIF(M,N)=ABS(DATA(M)-DATA(M+N))
0005              DO 10 I=1,4
0006            J=ISTART(I)
0007            K=1
0008            IF((I.EQ.1).OR.(I.EQ.3)) K=-1
0009            CTEMP=100.
0010            NTEMP=0
0011      1     J=J+K
0012            IF(((I.EQ.1).OR.(I.EQ.3)).AND.(J.LE.IEND(I))) GO TO 11
0013            IF(((I.EQ.2).OR.(I.EQ.4)).AND.(J.GE.IEND(I))) GO TO 11
0014            A=DIF(J,2)
0015            IF(A.GT.P) GO TO 1
0016            B=DIF(J,-2)
0017            IF(B.GT.P) GO TO 1
0018            C=DIF(J,4)
0019            IF(C.GT.Q) GO TO 2
0020            D=DIF(J,-4)
0021            IF(D.GT.Q) GO TO 2
0022            NCONT(I)=J
0023            CONT(I)=DSUM(J-4,J+4,DATA)/9.
0024            GO TO 10
0025      2     C=AMAX1(A,B)
0026            IF (NTEMP.EQ.0) GO TO 3
0027            IF((C-CTEMP).GT.3.) GO TO 1
0028              IF(DATA(J).LT.DATA(NTEMP)) GO TO 1
0029      3     CTEMP=C
0030            NTEMP = J
0031            GO TO 1
0032      11    IF (NTEMP.GT.0) GO TO 12
0033            NCONT(I)=0
0034            CONT(I)=0.0
0035            GO TO 10
0036      12    NCONT(I)=NTEMP
0037            CONT(I)=DSUM(NTEMP-2,NTEMP+2,DATA)/5.
0038      10    CONTINUE
0039              L1=NCENT/4+1
0040              L2=3*(NPTS-NCENT)/4+NCENT-1
0041              NP=NPTS-1
0042              M3=NSHIFT+1
0043              DO 30 J=1,M3,NSHIFT
0044              L1=L1+J-1
```

```
                L2=L2+J-1
                NP=NP+J-1
                K1=J+2
                K2=NP
                T1=0.
                T2=0.
      20        IF(DATA(K1).LT.T1) GO TO 22
                N1=K1
                T1=DATA(K1)
      22        IF(DATA(K2).LT.T2) GO TO 23
                N2=K2
                T2=DATA(K2)
      23        K1=K1+1
                K2=K2-1
                IF((K1.GT.L1).AND.(K2.LT.L2)) GO TO 24
                IF(K1.GT.L1) K1=L1
                IF(K2.LT.L2) K2=L2
                GO TO 20
      24        IF(J.GT.1) GO TO 25
                NCONT(1)=N1
                NCONT(2)=N2
                IF((CONT(1).NE.0.).OR.(CONT(2).NE.0.)) GO TO 30
                CONT(1)=0.333*(DATA(N1)+DATA(N1-1)+DATA(N1+1))
                CONT(2)=0.333*(DATA(N2)+DATA(N2-1)+DATA(N2+1))
                GO TO 30
      25        NCONT(3)=N1
                NCONT(4)=N2
                IF((CONT(3).NE.0.).OR.(CONT(4).NE.0.)) GO TO 30
                CONT(3)=0.333*(DATA(N1)+DATA(N1-1)+DATA(N1+1))
                CONT(4)=0.333*(DATA(N2)+DATA(N2-1)+DATA(N2+1))
      30        CONTINUE
              RETURN
              END

ROUTINES CALLED:
ABS    , DSUM   , AMAX1

SWITCHES = /ON

BLOCK        LENGTH
CONTNA  1009    (003742)*
STUFF   324     (001210)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES  00714 13077
EXECUTABLES   00927 12864
ASSEMBLY      01545 15163
```

```
                SUBROUTINE COMBIN(DATA,NPTS)
      C         FINDS CENTRAL WAVELENGTH OF PROFILE USING ZEROTH AND
      C         FIRST MOMENTS OF BOTH POLARIZATIONS
      C         CENTER IS AT K+PHD
      C         TDT 1/17/74
                DIMENSION DATA(512)
                EQUIVALENCE (K,LCENT)
                COMMON/STUFF/CONT(4),NCONT(4),COMT(2),ISTART(4),
              1 IEND(4),SUM(70),DIF(70),PHD,DELNC,SHIFT,NSHIFT,
              2 SCHIFF,NCENT,LCENT,SCALE,P,G,LIMIT,AREA
                ST=0.
                NIT=0
                NPT=NPTS+NSHIFT
                PHD=0.
          1     NIT=NIT+1
                LA=NCENT-NCONT(1)
                LB=IABS(NCENT-NCONT(2))
                LC=IABS(NCENT-NPTS+1)
                LIM=MIN0(LA,LB,LC)-2
                W=0.1/FLOAT(LIM)
                AM0=COSTR(DATA,512,NCENT,LIM,W,ST,PHD,1)
                AM1=ST/W
                NCEN=NCENT+NSHIFT
                LA=NCEN-NCONT(3)
                LB=IABS(NCEN-NCONT(4))
                LIM=MIN0(LA,LB,LC)-2
                W=0.1/FLOAT(LIM)
                BM0=COSTR(DATA,512,NCEN,LIM,W,ST,PHD,1)
                IF(ABS(AM0-BM0).GT.3000.) GO TO 11
                BM1=ST/W
                DELNC=0.5*(BM1/BM0+AM1/AM0)
                CENA=NCENT+PHD
                CEN=CENA+DELNC
         10     FORMAT (' OLD, NEW CENTERS ',2F10.2)
                TEST=CENA*0.25
                IF(ABS(DELNC).GT.TEST) GO TO 20
                IF(NIT.GT.10) GO TO 20
                NCENT=CEN+0.5
                K=NCENT
                PHD=CEN-FLOAT(NCENT)
                WRITE(5,169) AM0,BM0,DELNC,CENA,CEN,LIM
        169     FORMAT(2F20.5,3F7.3,I10)
                WRITE(6,169) AM0,BM0,DELNC,CENA,CEN,LIM
                IF((ABS(DELNC).LT.0.05).AND.(NIT.GT.1)) RETURN
                GO TO 1
         20     PRINT 11
         11     FORMAT (' SLOW CONVERGENCE IN COMBINE')
                CEN=CENA+0.5*DELNC
                K=CEN+0.5
                PHD=CEN-FLOAT(K)
                RETURN
                END
```

ROUTINES CALLED:
IABS , MIN0 , FLOAT , COSTR , ABS

SWITCHES = /ON

BLOCK LENGTH

```
FORTRAN V004A                           00:03:46     01-JAN-72   PAGE     1

0001             FUNCTION DSUM (N,M,DATA)
|         C   T. TARBELL    JULY 12, 1971.
0002               DIMENSION DATA(512)
0003             SUM=0.0
0004             IF(M.LT.N) GO TO 2
0005             IF ((N.LE.0).OR.(N.GT.512)) GO TO 2
0006             IF ((M.LE.0).OR.(M.GT.512)) GO TO 2
0007             DO 1 I=N,M
0008      1      SUM=SUM+DATA(I)
0009               DSUM=SUM
0010             RETURN
0011      2      DSUM=0.
0012             RETURN
0013             END

        SWITCHES = /ON

        BLOCK          LENGTH
        DSUM     140     (000430)*

        **COMPILER ----- CORE**
           PHASE         USED  FREE
        DECLARATIVES 00366 13425
        EXECUTABLES  00527 13264
        ASSEMBLY     00988 15720
```

```
      SUBROUTINE ZER(DAT,N,NC,LIMIT,SLOP,WNEW,PHD,IZERO,NCOS)
C   ZER RETURNS THE FIRST POSITIVE ZERO OF THE SINE TRANSFORM OF DAT
C   USES METHOD OF FALSE POSITION WITH 5 ITERATIONS
C   WNEW IS VALUE OF TRANS. VARIABLE AT ZERO.  IZERO=0 IF NO ZERO FOUND
C   ORIGIN IS AT NC, ENDPOINTS NC+LIMIT,NC-LIMIT.  LENGTH IS N
C   SLOPE IS THE SIN TRANS. AT W = .1/LIMIT   PHD IS PHASE FACTOR
C       IF NCOS LE 0, FINDS 1ST 2 ROOTS OF COS TRANS,
C       SLOPE=SMALLER ROOT, WNEW = LARGER ROOT
C   T. TARBELL   JULY 22, 1971
      DIMENSION DAT(N)
        SLOP=0.
        WNEW=0.
        IF(NCOS.LE.0) GO TO 41
      W2=0.1/FLOAT(LIMIT)
      A=COSTR(DAT,N,NC,LIMIT,  W2,SLOP,PHD,NCOS)
        SLOP=SLOP/(A*W2)
41      CONTINUE
        W2=0.075
        TEM1=TRAN1(DAT,N,NC,LIMIT,W2,PHD,NCOS)
        W1=W2
      S1=SIGN(1.,TEM1)
      TEM2=0.
1     W2=W2+0.025
      IF(W2.GE.1.) GO TO 4
      TEM2=TRAN1(DAT,N,NC,LIMIT,W2,PHD,NCOS)
      S2=SIGN(1.,TEM2)
      IF(S1.NE.S2) GO TO 20
      TEM1=TEM2
      S1=S2
      W1=W2
      GO TO 1
4     CONTINUE
      PRINT 100
100   FORMAT ('  NO ZERO FOUND')
      IZERO=0
      RETURN
20    CONTINUE
      TNEW=0.
      ITER=0
21    ITER=ITER+1
      WNEW=W1-(W2-W1)*TEM1/(TEM2-TEM1)
        IF(ITER.GT.4) GO TO 30
        TNEW=TRAN1(DAT,N,NC,LIMIT,WNEW,PHD,NCOS)
      SNEW=SIGN(1.,TNEW)
      IF(SNEW.EQ.S1) GO TO 22
      S2=SNEW
      TEM2=TNEW
      W2=WNEW
      GO TO 21
22    S1=SNEW
      TEM1=TNEW
      W1=WNEW
      GO TO 21
30    CONTINUE
        IF(NCOS.GT.0) RETURN
```

```
      IF(SLOP.NE.Ø.) RETURN
      SLOP=WNEW
      W1=AMAX1(W1,W2)
      IF(W1.NE.W2) GO TO 43
      S1=S2
      TEM1=TEM2
      GO TO 1
43    W2=W1
      GO TO 1
      END
```

```
ROUTINES CALLED:
FLOAT , COSTR , TRAN1 , SIGN   , AMAX1

SWITCHES = /ON

BLOCK        LENGTH
ZER      615     (ØØ2316)*

**COMPILER -----  CORE**
   PHASE         USED  FREE
DECLARATIVES   ØØ366 13425
EXECUTABLES    ØØ687 13104
ASSEMBLY       Ø1364 15344
```

```
      FUNCTION COSTR(DATA,N,NC,LIMIT,W,SINTR,PHD,NCOS)
C   RETURNS THE COSINE TRANSFORM (COSTR) AND SINE TRANSFORM (SINTR
C   USES SINCOS RECURSION FORMULAS FOR SPEED. NOT AN FFT. DONT INVE
C   DATA IS THE ARRAY OF DIMENSION N, WITH ORIGIN AT NC AND ENDPOINTS
C   AT +-LIMIT.   W IS VALUE OF TRANSFORM VARIABLE.  +SIN(W*X) IS USED
C   THE TRANSFORM IS THEN SHIFTED IN PHASE BY -PHD
C   T. TARBELL JULY 22, 1971
C       IF NCOS NEG., COSTR=SIN. TRAN. SINTR=COS. TRAN.
      DIMENSION DATA(N)
      CT=DATA(NC)
      IF(W.EQ.0.) GO TO 10
      ST=0.
      CTEM=1.0
      STEM=0.
      COSW=COS(W)
      COSW2=2.*COSW
      SINW=SIN(W)
      DO 1 I=1,LIMIT
      J=NC+I
      K=NC-I
      CT=CT+COSW*(DATA(J)+DATA(K))
      ST=ST+SINW*(DATA(J)-DATA(K))
      H=COSW2*COSW-CTEM
      CTEM=COSW
      COSW=H
      H=COSW2*SINW-STEM
      STEM=SINW
    3 SINW=H
    1 CONTINUE
      IF(PHD.NE.0.) GO TO 2
      SINTR=ST
      COSTR=CT
        IF(NCOS.GE.0)RETURN
        T=SINTR
        SINTR=COSTR
        COSTR=T
      RETURN
    2 PH=PHD*W
      CPH=COS(PH)
      SPH=SIN(PH)
      SINTR=CPH*ST-SPH*CT
      COSTR=CPH*CT+SPH*ST
      RETURN
   10 DO 11 I=1,LIMIT
   11 CT=CT+DATA(NC+I)+DATA(NC-I)
      COSTR=CT
      SINTR=0.
        IF(NCOS.GE.0) RETURN
        SINTR=COSTR
        COSTR=0.
      RETURN
      END

ROUTINES CALLED:
COS    , SIN

SWITCHES = /ON

BLOCK        LENGTH
```

```
              FUNCTION TRAN1(DATA,N,NC,LIMIT,W,PHD,NCOS)
       C      SAME AS COSTR, BUT COMPUTES ONLY ONE TRANSFORM
       C      TRAN1 = SIN. TRAN. IF NCOS.GT.0
       C            = COS. TRAN. IF NCOS.LE.0
       C      TDT  4/5/74
              DIMENSION DATA(N)
              IF(NCOS.LE.0) GO TO 200
              PH=W*PHD
              STEM=-SIN(PH)
              ST=DATA(NC)*STEM
              COSW=COS(W)
              SINW=SIN(W)
              COSW2=2.*COSW
              DO 100 I=1,LIMIT
              J=NC+I
              K=NC-I
              ST=ST+SINW*(DATA(J)-DATA(K))
              H=COSW2*SINW-STEM
              STEM=SINW
              SINW=H
       100    CONTINUE
              TRAN1=ST
              RETURN
       200    CONTINUE
       C      COS. TRAN. PART NOT NEEDED YET
              TRAN1=0.
              RETURN
              END
```

```
ROUTINES CALLED:
SIN    , COS

SWITCHES = /ON

BLOCK        LENGTH
TRAN1    263    (001016)*

**COMPILER --*-- CORE**
   PHASE       USED  FREE
DECLARATIVES  00446 13345
EXECUTABLES   00607 13184
ASSEMBLY      01084 15624
```

```
              SUBROUTINE POLYAN(DATA,N,NC,LIMIT,PHD,KMODE)
      C       OVERSEES FOURIER TRANSFORM AND LEG. POL.
      C       ANALYSIS OF CIRCULAR POARIZATIONS
      C       IF ABS(COSGAM) GT Ø.7, ALSO USES SEARS RELS. ON B/(A-C)
              DIMENSION DATA(N),TX(9),GPT(5),PLCOF(5),GWT(5,5),SPT(5)
              COMMON/RESLTS/X,COSGAM,GAM,SLOPE,IZERO,NORAT,
            1 PLCOF,A,B,C,ALOR
              DATA SPT/1.Ø,Ø.9139Ø,Ø.67Ø95,Ø.3149Ø,Ø./
              DATA NGPT/4/,GPT/1.57Ø8,1.1529,Ø.7355,Ø.32Ø4,
            1 Ø./, GWT/ .2Ø89795Ø, Ø., -.52244898, Ø., .7Ø53Ø612,
            2 .19Ø915Ø3, .23244581, -.241445Ø5, -.59Ø224Ø8, -.213Ø1324,
            3 .13985270, .31111541, .2271227Ø, -.Ø9Ø98Ø65, -.45841186,
            4 .Ø6474248, .18434281, .27554684, .32346645, .318772Ø4,
            5 5*Ø./
              DATA PI/3.14159265/,RAD/57.2958/
              ACOS(Y)=ASIN(SQRT(1.-Y*Y))
              IZERO=5
              CALL ZER(DATA,N,NC,LIMIT,SLOPE,WNEW,PHD,IZERO,1)
              IF(KMODE.EQ.1) SLOPE=-SLOPE
              IF(IZERO.EQ.Ø) RETURN
              X=PI/WNEW
              DO 1Ø M=1,NGPT
              TX(M)=RAT(DATA,N,NC,LIMIT,PHD,GPT(M),SPT(M),X,NORAT)
              IF(NORAT.NE.Ø) RETURN
      1Ø      CONTINUE
              DO 11 M=2,NGPT
              CKX=PI-GPT(M)
              TX(M+NGPT-1)=RAT(DATA,N,NC,LIMIT,PHD,CKX,SPT(M)
            1  ,X,NORAT)
              IF(NORAT.NE.Ø) RETURN
      11      CONTINUE
              CALL PLINT(TX,PLCOF,NGPT,GWT)
              FAC=1./(PLCOF(1)+PLCOF(2))
              B=PLCOF(1)*FAC
              A=Ø.5*(1.+PLCOF(2))*FAC
              C=A-FAC
              IF(ABS(PLCOF(2)).GT.1.) GO TO 11Ø
              COSGAM=1.*SIGN(1.,PLCOF(2))
              GO TO 199
      11Ø     COSGAM=PLCOF(2)-SIGN(1.,PLCOF(2))*SQRT(PLCOF(2)*PLCOF(2)
            1 -1.)
      199     IF(KMODE.EQ.1) COSGAM=-COSGAM
      2ØØ     CONTINUE
              RETURN
              END

ROUTINES CALLED:
ASIN    , SQRT    , ZER    , RAT    , PLINT  , ABS    , SIGN


SWITCHES = /ON


BLOCK         LENGTH
POLYAN  623     (ØØ2336)*
RESLTS  28      (ØØØØ7Ø)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES ØØ366 13425
```

```
              FUNCTION RAT(DATA,N,NC,LIMIT,PHD,PHAS,SKX,X,NORAT)
C             RAT = SIN(KX) * COSINE TRANS(K) / SIN TRANS(K)
C             SHOULD LOOK LIKE C1 + C2 * COS(KX)
C             NORAT.NE.Ø IF INFINITY ENCOUNTERED
              DIMENSION DATA(N)
              DATA SMALL/1.E-4/
              NORAT=Ø
              W=PHAS/X
              CT=COSTR(DATA,N,NC,LIMIT,W,ST,PHD,1)
              SMAL=SMALL*ABS(CT)
              IF(ABS(ST).GT.SMAL) GO TO 1Ø
              NORAT=1Ø
              RAT=1.
              RETURN
  1Ø          CONTINUE
              RAT=CT*SKX/ST
              RETURN
              END

      ROUTINES CALLED:
      COSTR , ABS

      SWITCHES = /ON

      BLOCK        LENGTH
      RAT      183    (ØØØ556)*

      **COMPILER --*-- CORE**
         PHASE          USED  FREE
      DECLARATIVES    ØØ366 13425
      EXECUTABLES     ØØ6Ø7 13184
      ASSEMBLY        Ø1Ø2Ø 15688
```

```
FORTRAN V004A                    01:18:23    09-APR-74    PAGE    1

                 SUBROUTINE PLINT(TX,PLCOF,NGPT,GWT)
        C     PLCOF(NLEG) = COEFF. OF (NLEG-1)TH LEGENDRE POLY. IN
        C       EXPANSION OF TX(COSKX)
        C       GAUSSIAN QUAD.  2*NGPT-1 PTS.
                 DIMENSION TX(9),PLCOF(5),GWT(5,5)
                 DO 10 NLEG=1,5
                 SUM=0.
                 DO 1 I=1,NGPT
        1        SUM=SUM+TX(I)*GWT(NLEG,I)
                 SGN=1.
                 IF((NLEG.EQ.2).OR.(NLEG.EQ.4)) SGN=-1.
                 NL=NGPT+1
                 NU=NGPT+NGPT-1
                 DO 2 I=NL,NU
        2        SUM=SUM+SGN*TX(I)*GWT(NLEG,I-NGPT+1)
        10       PLCOF(NLEG)=SUM
                 RETURN
                 END

        SWITCHES = /ON

        BLOCK        LENGTH
        PLINT    214     (000654)*

        **COMPILER ----- CORE**
           PHASE         USED  FREE
        DECLARATIVES  00446 13345
        EXECUTABLES   00527 13264
        ASSEMBLY      01044 15664
```

```
      SUBROUTINE BGOOD(GOLD,B,GAMF)
      DIMENSION GOLD(2,8)
      DATA DXMAX,DMMAX,RESMAX,AVMIN/.25,.15,.15,150./
      PERCEN(X,Y)=2.*ABS(X-Y)/(X+Y+1.E-4)
      ACOS(Y)=ASIN(SQRT(1.-Y*Y))
      DATA PI,RAD/3.1415927,57.2958/
      DM=PERCEN(GOLD(1,4),GOLD(2,4))
      IF(DM.GT.DMMAX) GO TO 100
      AVMOM=0.5*(GOLD(1,4)+GOLD(2,4))
      IF(ABS(AVMOM).LT.AVMIN) GO TO 101
      ILOOK=1
      IF(GOLD(2,7).GT.RESMAX) GO TO 50
      ILOOK=2
      IF(GOLD(1,7).GT.RESMAX) GO TO 60
      GO TO 51
50    IF(GOLD(1,7).LT.RESMAX) GO TO 60
51    W1=1./PERCEN(GOLD(1,4),GOLD(1,5))
      W2=1./PERCEN(GOLD(2,4),GOLD(2,5))
      T=1./(W1+W2)
      W1=W1*T
      W2=W2*T
      B=W1*GOLD(1,2)+W2*GOLD(2,2)
      CGAMF=W1*GOLD(1,3)+W2*GOLD(2,3)
      GAMF=ACOS(CGAMF)
      IF(CGAMF.LE.0.) GAMF=PI-GAMF
      GAMF=GAMF*RAD
      RETURN
60    B=GOLD(ILOOK,2)
      GAMF=ACOS(GOLD(ILOOK,3))
      IF(GOLD(ILOOK,3).LE.0.) GAMF=PI-GAMF
      GAMF=GAMF*RAD
      RETURN
100   B=0.
      GAMF=0.
      RETURN
101   B=ABS(AVMOM)
      GAMF=0.
      IF(AVMOM.LE.0.) GAMF=180.
      RETURN
      END
```

```
ROUTINES CALLED:
ABS    , ASIN   , SQRT

SWITCHES = /ON

BLOCK         LENGTH
BGOOD    594    (002244)*

**COMPILER -----  CORE**
   PHASE        USED  FREE
DECLARATIVES  00366 13425
EXECUTABLES   00687 13104
ASSEMBLY      01264 15444
```

```
C         PROGRAM MGRAM
C         READS PASS3 OUTPUT TAPE AND PRINTS MAPS OF RESULTS
C         NEEDS SUBROUTINES PACK, PMAPS
C         HANDLES ANY NO. PTS. ON SUN, PRINTED OUT IN BLOCKS
C         OF NMAX...TO CHANGE NMAX, CHANGE DIM.,EQ. & DATA
C         STATEMENTS FOR R, B, V, GAM, AND NMAX
C         TDT  APRIL 5, 1974
          DIMENSION IFIL(5),R(4500), B(1500),GAM(1500),V(1500),
     1    TITLE(3,5)
          COMMON IDATA(100),MAPS(5),DELX,DELY
          EQUIVALENCE (B(1),R(1)),(GAM(1),R(1501)),(V(1),
     1    R(3001)),(XSTEP,IDATA(1)),(YSTEP,IDATA(3)),(NX,
     2    IDATA(5)),(NY,IDATA(6))
          DATA R/4500*0./,NMAX/1500/
          DATA IBLANK/'  '/
          DATA TITLE/'B ',2*'  ','GA','M ','  ','V ',2*'  ',
     1    'CO','NT','  ','BV','ER','T '/
    5     WRITE (6,1010)
 1010     FORMAT(' TYPE FILENAME .CR TO ABORT'/)
          READ (6,1011) IFIL
 1011     FORMAT (5A2)
          IF(IFIL(1).EQ.IBLANK) GO TO 200
          WRITE (6,700)
  700     FORMAT(/' WHICH MAPS SHALL MOUSE PRINT?'/
     1    ' TYPE 1 FOR YES, 0 FOR NO (I1 FORMAT)'/)
          DO 710 I=1,5
          WRITE (6,701) (TITLE(J,I),J=1,3)
  701     FORMAT(/'    ',3A2/)
          READ(6,702) MAPS(I)
  702     FORMAT(I1)
  710     CONTINUE
          CALL PFEED(0)
          PRINT 1012,IFIL
 1012     FORMAT (//'  FILENAME  ',5A2//)
          CALL SETFIL(8,IFIL)
    1     READ(8,END=100) ICODE,M3,(IDATA(M13),M13=1,M3)
    2     ICODE=ICODE+1
          GO TO (50,10,20) ICODE
   10     PRINT 1000,(IDATA(M),M=1,M3)
          WRITE(6,1000) (IDATA(M),M=1,M3)
 1000     FORMAT (500A2)
          GO TO 1
   20     NXTOT=NX
          NYTOT=NY
          DELX=XSTEP
          DELY=YSTEP
   21     IOVFLO=0
          IF(NXTOT*NYTOT.LE.NMAX) GO TO 22
          NY1ST=NYTOT
          IOVFLO=1
          NYTOT=INT(NMAX/NXTOT)
   22     CONTINUE
          PRINT 1102,NXTOT,NYTOT,DELX,DELY
 1102     FORMAT(/' NX, NY, DELX, DELY'2I5,2F7.3/)
          GO TO 1
```

```
      50   CALL PACK(R,NXTOT,NYTOT,B,GAM,V)
           IF(IOVFLO.EQ.0) GO TO 5
           NYTOT=NY1ST-NYTOT
           GO TO 21
      100  END FILE 8
           PRINT 1004
           WRITE(6,1004)
      1004 FORMAT(/'  EOF FOUND BY MAIN PROGRAM',/)
           GO TO 5
      200  PRINT 1013
           WRITE(6,1013)
      1013 FORMAT(/'  RUN ENDED')
           STOP
           END

ROUTINES CALLED:
PFEED , SETFIL, INT    , PACK

SWITCHES = /ON

BLOCK        LENGTH
MAIN.    9738    (046024)*
.$$$$.   109     (000332)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES  00683 13108
EXECUTABLES   00767 13024
ASSEMBLY      01593 15115
```

```
FORTRAN V004A                     00:26:46     09-APR-74    PAGE     1

              SUBROUTINE PACK(R,NX,NY,B,GAM,V)
              COMMON IDATA(100),MAPS(5),DELX,DELY
              DIMENSION R(2),JDATA(10),B(NX,NY),V(NX,NY),
           1  GAM(NX,NY)
              EQUIVALENCE (BEE,IDATA(5)),(GAMA,IDATA(7)),
           1  (VEE,IDATA(3)),(IX,IDATA(1)),(IY,IDATA(2))
              EQUIVALENCE (BE,JDATA(5)),(GAMM,JDATA(7)),
           1  (VE,JDATA(3)),(JX,JDATA(1)),(JY,JDATA(2))
              B(IX,IY)=BEE
              GAM(IX,IY)=GAMA
              V(IX,IY)=VEE
      1       READ (8,END=100) ICODE,M3,(JDATA(M33),M33=1,M3)
              IF(ICODE.NE.0) GO TO 200
              B(JX,JY)=BE
              V(JX,JY)=VE
              GAM(JX,JY)=GAMM
              IF((JX.NE.NX).OR.(JY.NE.NY)) GO TO 1
              CALL PMAPS(B,GAM,V,NX,NY)
              RETURN
      100     PRINT 1000
              WRITE (6,1000)
      1000    FORMAT(/' PACK FOUND EOF'/)
      101     B(JX,JY)=BE
              V(JX,JY)=VE
              GAM(JX,JY)=GAMM
              CALL PMAPS(B,GAM,V,NX,JY)
              RETURN
      200     PRINT 1001, ICODE
              WRITE (6,1001) ICODE
      1001    FORMAT(/' DATA EXPECTED..ICODE ='I5/)
              IF(ICODE.EQ.1) WRITE (6,1002) JDATA
              IF(ICODE.EQ.1) PRINT 1002,JDATA
      1002    FORMAT(/' '40A2/)
              CALL PMAPS(B,GAM,V,NX,JY)
              RETURN
              END

      ROUTINES CALLED:
      PMAPS

      SWITCHES = /ON

      BLOCK        LENGTH
      PACK     485     (001712)*
      .$$$$.   109     (000332)

      **COMPILER ----- CORE**
         PHASE        USED  FREE
      DECLARATIVES 00366 13425
      EXECUTABLES  00793 12998
      ASSEMBLY     01361 15347
```

```
                SUBROUTINE PMAPS(B,GAM,V,NX,NY)
                COMMON IDATA(100),MAPS(5),DELX,DELY
                DIMENSION SPACE(30)
                EQUIVALENCE (SPACE(1),IDATA(1))
                DIMENSION B(NX,NY),V(NX,NY),GAM(NX,NY)
                DATA RAD/57.2958/
                DO 100 I=1,5
                IF(MAPS(I)) 1,100,1
      1         GO TO (10,20,30,40,50) I
      10        CALL PFEED(0)
                PRINT 101,DELX,DELY
      101       FORMAT(/' MAG. FIELD (GAUSS) ON 'F7.3' BY '
             1  F7.3' ARCSECOND GRID'//)
                DO 11 K=1,NY
      11        PRINT 102,(B(L,K),L=1,NX)
      102       FORMAT(/25F5.0)
                GO TO 100
      20        CALL PFEED(0)
                PRINT 201,DELX,DELY
      201       FORMAT(/' GAMMA (DEGREES) ON 'F7.3' BY 'F7.3
             1  ' ARCSECOND GRID'//)
                DO 21 K=1,NY
      21        PRINT 102,(GAM(L,K),L=1,NX)
                GO TO 100
      30        CALL PFEED(0)
                PRINT 301,DELX,DELY
      301       FORMAT(/' REL. VEL. (KM/SEC: + = BLUESHIFT) ON 'F7.3
             1  ' BY 'F7.3' ARCSECOND GRID'//)
                DO 31 K=1,NY
      31        PRINT 302,(V(L,K),L=1,NX)
      302       FORMAT(/25F5.1)
                GO TO 100
      40        CALL PFEED(0)
                PRINT 401,DELX,DELY
      401       FORMAT(/' CONTINUUM INTENSITY ON 'F7.3' BY '
             1  F7.3' ARCSECOND GRID'//)
                DO 41 K=1,NY
                DO 42 L=1,NX
      42        SPACE(L)=100.*(B(L,K)/10.-INT(B(L,K)/10.))
                PRINT 102,(SPACE(L),L=1,NX)
      41        CONTINUE
                GO TO 100
      50        CALL PFEED(0)
                PRINT 501,DELX,DELY
      501       FORMAT(/' VERT. MAG. FIELD (GAUSS) ON 'F7.3
             1  ' BY 'F7.3' ARCSECOND GRID'//)
                DO 51 K=1,NY
                DO 52 L=1,NX
      52        SPACE(L)=B(L,K)*COS(GAM(L,K)/RAD)
                PRINT 102,(SPACE(L),L=1,NX)
      51        CONTINUE
      100       CONTINUE
                RETURN
                END
```

ROUTINES CALLED:
PFEED , INT , COS

SWITCHES = /ON